Universität des Saarlandes

Modularity and Determinism in Compositional Markov Models

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurswissenschaften (Dr.Ing)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Pepijn Crouzen

Saarbrücken
März 2014

Kolloquium	14.08.2014
Promotionsausschuss:	
1. Gutachter	Prof. Dr. Holger Hermanns
2. Gutachter	Dr. Mariëlle I.A. Stoelinga
3. Gutachter	Dr. Pedro R. D'Argenio
Vorsitzender und Dekan	Univ.-Prof. Dr. Markus Bläser

Acknowledgements

First and foremost, I'd like to thank my advisor, Holger Hermanns, for his support during my time at Saarland University. I have learned a tremendous amount (and not just about Markov chains). I'd also like to thank Mariëlle Stoelinga and Pedro D'Argenio for the time they took to read this thesis and the critical comments they made that helped me improve it.

I also want to thank all my coworkers at Saarland University for the wonderful time working together, and especially my roommate Reza Pulungan for all the interesting debates and discussions.

I'm also very grateful to my family for their love and support. In particular, I have to thank Verena Wolf, since without her this thesis would never have been completed. And last but not least I'd like to thank my daughter Heleen and my son Adriaan, just because.

Summary

Markov chains are a versatile and widely used means to model an extensive variety of stochastic phenomena, but describing a complex system as a monolithic Markov chain is difficult and error-prone. In this thesis we show that we can construct such complex Markov chains in a sound manner through the composition of a number of simple input/output interactive Markov chains (I/O-IMCs), which arise as an orthogonal combination of continuous-time Markov chains and input/output automata).

I/O-IMCs come equipped with a modular semantics in terms of interactive jump processes, a novel variation of jump processes. We discuss the phenomenon of non-determinism, arising from the interaction inside such models, and how we can efficiently determine whether a complex I/O-IMC model is deterministic. Finally, we give an example of an application of I/O-IMCs by presenting the ARCADE language, which can be used to describe complex dependable systems.

In this thesis we show that, by providing a modular semantics for our compositional I/O-IMCs, we achieve the 'triple compositionality' principal: a simple, but powerful compositional syntax (ARCADE), has an interactive and Markovian semantics in terms of I/O-IMCs, which gives an intuitive description of the meaning of each syntactic element. I/O-IMCs themselves then have a stochastic semantics in terms of interactive jump processes which enables us to describe and compute their stochastic properties. This triple compositionality provides a natural, non-monolithic semantics for our high-level syntax and allows us to understand and reason about complex, incomplete, or partially-specified stochastic models.

Zusammenfassung

Markov-Ketten sind ein vielseitiges und weit verbreitetes Mittel zur Modellierung einer Vielzahl von stochastischen Phänomenen, aber es ist schwierig und fehleranfällig, ein komplexes System als monolithische Markov-Kette zu beschreiben. In dieser Arbeit zeigen wir, dass solche komplexen Markov-Ketten auf korrekte Weise durch die Komposition einer Anzahl von einfachen input/output interactive Markov chains (I/O-IMCs), die als orthogonale Kombination von zeitkontinuierlichen Markov-Ketten und input/output automata zustande kommen, konstruiert werden können.

I/O-IMCs sind ausgestattet mit einer modularen Semantik in der Form von interaktiven Sprungprozessen, einer neuartigen Variante von Sprungprozessen. Weiterhin diskutieren wir das Phänomen des Nicht-Determinismus, der sich aus der Interaktion innerhalb solcher Modelle ergibt, und wie wir effizient bestimmen können, ob ein komplexes I/O-IMC Modell deterministisch ist. Schließlich geben wir ein Beispiel fr eine Anwendung von I/O-IMCs: die ARCADE Sprache, die verwendet werden kann, um komplexe zuverlässige Systeme zu beschreiben.

In dieser Arbeit zeigen wir, dass wir durch die Beschreibung einer modularen Semantik für unsere I/O-IMCs das 'Triple-Compositionality-Prinzip' erreichen: eine einfache, aber leistungsfhige kompositionelle Syntax (ARCADE), hat eine interaktive und markovsche Semantik in Form von I/O-IMCs, die eine intuitive Beschreibung der Bedeutung der einzelnen syntaktischen Elementen darstellt. I/O-IMCshaben außerdem eine stochastische Semantik in Form von interaktiven Sprungprozessen, die es ermöglicht, ihre stochastischen Eigenschaften zu beschreiben und zu berechnen. Dieses 'Triple-Compositionality-Prinzip' bietet eine natrliche nicht-monolithische Semantik und erlaubt es, komplexe, unvollständige oder unterspezifierte stochastiche Modelle zu verstehen und zu beschreiben.

Contents

1 Introduction

Markov chains are a versatile and widely used means to model an extensive variety of stochastic phenomena from bio-chemical reaction networks [19] to the performance of computer systems [22], and even the structure of the internet [40]. Besides their versatility, one of the main reasons why Markov chains are so popular is their simplicity. Under mild assumptions, the dynamics of a Markov chain can be represented by a simple matrix. This makes Markov chains easy to represent and analyse using linear-algebraic techniques.

This thesis studies a way to model and analyse complex stochastic systems in a *compositional* fashion. To illustrate the main innovation, consider the following scenario: we want to study the reliability of a cooling system consisting of several pumps, valves, and filters. Each of the components of such a system behaves stochastically: after some random delay, a pump or a valve of the system may fail. We may be able to model the behaviour of these components of the system using a stochastic model (such as a Markov chain). To study the stochastic behaviour of the entire cooling system we need some way of combining the representations of the components (pumps, valves, and filters) to define the representation of the whole system. *Input/output interactive Markov chains* (I/O-IMCs) enables us to do exactly that. The formalism of I/O-IMCs allows us to describe both the stochastic aspect of the pumps, valves, and filters and the way in which these components interact. A description of the entire system then arises naturally through the *composition* of its components. To see how we arrive at the formalism of I/O-IMCs we will first discuss Markov chains.

1.1 Markov chains

Markov chains come in two flavours depending on the way time is modelled: discrete-time Markov chains (DTMCs), where time proceeds in discrete steps, and continuous-

time Markov chains (CTMCs), where time is continuous. In general, DTMCs are useful for modelling systems where all events occur synchronously, whereas CTMCs are more appropriate for modelling systems where different events may occur on different time scales. In this thesis we will use CTMCs as the way to model stochastic phenomena and we will focus on systems that exhibit widely different time-scales. For instance, in a dependable system, the time it takes to repair some component of the system is usually several orders of magnitude smaller than the mean time between failures of that component.

1.1.1 Syntax and semantics

It is very common to represent a CTMC as a graph with positive real numbers on the edges. We can induce a Markov chain from such a graph by giving it a very simple semantics: the graph represents a *jump process*, namely a stochastic process with a state space given by the set of vertices of the graph and where the probability of jumping from a state x to a state y in an infinitesimal time interval is proportional to the real number on the edge from x to y. Moreover, this probability is independent of any past jumps of the jump process. It is well-known that this simple assumption is enough to construct the complete Markov chain from the graph [1] and we will revisit this construction in Chapter 3. We could say that the graph is the syntax or description of the CTMC, whereas the CTMC viewed as the above jump process is the semantics or meaning of the graph. In matrix form, the graph is usually referred to as the infinitesimal generator matrix of the chain.

1.1.2 Composition

Let us return to our example of a dependable cooling system consisting of a number of different components. We might hope to model such a system directly as a Markov chain, but unfortunately the size of this Markov chain would grow excessively with the number of components of the system we need to consider. Creating such a large, monolithic description of a complex system is thus both difficult and error-prone. Instead, it would be much simpler if we were able to model the components of the system individually and then combine them to thereby obtain a faithful model of the entire system. If the combination of components is a generic operation, this ensures that the modelling effort grows only linearly in the number of components of the system.

This may raise the question whether we can compose different CTMCs to build up complex CTMCs? In fact, there is a direct way of accomplishing this if one assumes that various components of the system are independent. In terms of the graphs that represents the component CTMCs we can indeed construct their "composition" by *interleaving* the edges of the two component graphs (see for instance [26]). Equivalently, we can construct the "composition" of two infinitesimal generator matrices by taking their Kroenecker product [46]. This results in a faithful representation of the composed system and its associated jump process, and in fact this construction is entirely *modular*: What we get is the jump process of the two composed graphs; by assuming the two

underlying jump processes are independent we obtain their cross-product jump process. In other words, we can find the semantics of the composition of two CTMCs from the semantics of the component CTMCs, and this semantic composition exactly matches the syntactic composition on the graph representation of CTMCs. We will use this simple notion of composition for CTMCs as one of the ingredients in our compositional Markovian model.

Unfortunately, this way of composing Markov chains is very uninteresting. Our assumption that the component CTMCs are independent, means that the corresponding components of the system do not influence each other, so we are bound to complex system models where components are completely isolated. But for those systems we can model and study the components in isolation. There is no point in composing them in the first place, if they do not interact in some way or another. What we are missing is a way to represent the fact that components in a system may indeed *interact*. To put it differently, we need a way to model that the behaviour of one component in a system may be influenced (changed) by the behaviour of other components in the same system. In this thesis, we will attack this problem by combining CTMCs with a purely interaction-oriented discrete-state formalism.

1.2 IOA

The input/output automata model (IOA) was introduced by Lynch and Tuttle to study distributed algorithms [33]. They describe distributed algorithms by modelling their component algorithms, thus avoiding the need to give a monolithic description for the entire distributed algorithm. In fact, the use of IOA not only allows for the modelling of complex distributed algorithms, it also turns out to be a great aid in the analysis of them. Lynch and Tuttle showed that, under certain fairness assumptions, properties of the distributed algorithm can be derived directly from the properties of its components [33]. We will give a short sketch of IOA here and we will review the theory of IOA in detail in Chapter 4.

1.2.1 Syntax and semantics

As we did for CTMCs, we will treat IOA as a graph-based formalism. In essence, an IOA is a graph where vertices represent states and the edges are labelled with *actions*. Actions represent different types of "events" that may happen. An example of a type of event is "button is pushed" or "message X is sent". The intuitive meaning of an edge, also called a transition, from vertex x to vertex y with action a is that, when the component is in state x and an event of type a happens, then the component will change to state y. We will dive deeper into the details of IOA in Chapter 4.

We have seen that a transition of an IOA represents the occurrence of an event of a particular type. The semantics of an IOA is then the set of possible sequences of actions that may happen. Such a sequence of action is called a trace. It should be mentioned that certain sequences of actions are considered *unfair* and are therefore not considered. The use of such *fairness assumptions* makes sure that certain unrealistic or undesirable

phenomena are not possible. An example of undesired behaviour is when actions of one component in a distributed algorithm are indefinitely postponed by other components' actions. The set of fair traces of an IOA represents all the different sequences of actions that may occur for that IOA.

1.2.2 Composition

IOA allow us to model the way in which components *interact*. We say two components interact, if the behaviour of one component influences the behaviour of the other. That is, the semantics of an IOA is different in the context of another IOA, than its semantics in isolation. In terms of the graph-syntax of an IOA, interaction is modelled by *synchronising* the transitions of IOA that we wish to compose. In essence, when we compose two IOA, any pair of transitions that have the same action label must happen at the same time. Pairs of transitions with different action labels are interleaved.

In this way, we can construct, given two graphs representing IOA, a graph representing their composition. The semantics of such a composite IOA is again its set of fair traces. Lynch and Tuttle have shown that composition for IOA is *sound*: when we project a fair trace of a composite IOA onto its components we obtain fair traces of the component IOA [33]. This allows us to prove properties of complex distributed algorithms by studying their components: if a sequence of actions if *not* a fair trace of the IOA representing a component of the algorithm, then this sequence of actions will be impossible in the complete distributed algorithm.

We have seen that both CTMCs and IOA have a sound compositional semantics. In this thesis, we will combine these two formalisms in an orthogonal way to find a compositional Markov model which again has a sound compositional semantics.

1.3 I/O-IMCs

IOA thus give us a way to model interaction between components, whereas in CTMCs we found a way to model stochastic phenomena. We can combine CTMCs and IOA to find a compositional way of modelling complex stochastic systems: input/output interactive Markov chains (I/O-IMCs) are designed to be used to model and analyse complex stochastic systems in a compositional way. The representation of an I/O-IMC is a graph whose edges are labelled with either positive real values *or* actions. We call the former *Markovian* transitions and the latter *interactive* transitions. The composition operator is also an orthogonal combination of composition for CTMCs and IOA: Markovian transitions, as well as transitions with different actions are interleaved, while transitions where the actions have equal names are synchronised. We will dive into the details of the graph representation of I/O-IMCs in Chapter 5.

1.3.1 A compositional semantics for I/O-IMCs

In order to use I/O-IMCs to represent the components of complex stochastic systems, it is important to understand the semantics of an I/O-IMC model in isolation, and

in composition. In Chapter 6 we will equip I/O-IMCs with a semantics, orthogonally combining the semantics of CTMCs and IOA: Markovian transitions will occur stochastically as for CTMCs and interactions will occur according to the semantics of IOA. As a semantical underpinning of I/O-IMCs we introduce *interactive jump processes* a variation on classical jump processes. This is the first modular semantics for I/O-IMCs. A core insight is that composition of I/O-IMCs is again *sound* with respect to its semantics. It is important to note that the semantics of an I/O-IMC is non-deterministic: an I/O-IMC is represented by a set of interactive jump processes, in the same way as an IOA is represented by a set of fair traces.

We can thus proceed and employ I/O-IMCs to give semantics to a complex dependable system, by modelling its components with I/O-IMCs. The I/O-IMC semantics of the entire system arises naturally by composing the I/O-IMCs that represent its components.

Still, being able to model complex dependable systems is not the end of the story. We also wish to analyse such systems, to quantify different dependability properties. In Chapter 7 we will study the semantics of *closed* I/O-IMCs, i.e., I/O-IMCs that do interact among their components, but do not interact with the surrounding environment. Such I/O-IMCs arise as models of complete dependable system and generally are the result of composition of many component I/O-IMCs. We will see that we can translate closed I/O-IMCs into continuous-time Markov decision processes (CTMDPs) [28]. CTMDPs can be seen as extensions of CTMCs where, at each state, many probabilistic transitions are possible; the choice between these transitions is performed in a non-deterministic fashion. The translation allows us to apply standard CTMDP analysis techniques to analyse closed I/O-IMCs.

1.3.2 Determinism

Initially, we started off with the intention to provide a way to construct stochastic models in a compositional way. However, the semantics of an I/O-IMC is in general not a stochastic process, but rather a stochastic non-deterministic model, such as a CTMDP. This is also the case for the particular instance of closed I/O-IMCs i.e., I/O-IMCs that do not interact with their environment. It is caused by the fact that IOA are inherently non-deterministic. The semantics of an IOA is a set of fair traces, representing a non-deterministic selection of the possible fair traces of the model. This non-determinism is inherited by I/O-IMCs, where the semantics gives rise to sets of interactive jump processes. However, not all I/O-IMCs are non-deterministic. Certain I/O-IMCs in fact do not contain non-deterministic choices. In Chapter 7 we will establish that the semantics of such a *deterministic* I/O-IMC is indeed a single interactive jump process, thus a CTMC.

When the semantics of a closed deterministic I/O-IMC is a CTMC, this allows us to apply standard Markov chain solution techniques to analyse such I/O-IMCs. This is of interest since CTMC analysis techniques are very well established and generally more efficient than CTMDP analysis techniques, where efficiency and analysis improvements are still subject of ongoing research [38, 9]. In addition, the absence of non-determinism

makes it possible to apply an entirely different class of algorithms for CTMCs, namely those that can be applied *on-the-fly*, i.e. without constructing the entire graph of the CTMC prior to analysis. For deterministic I/O-IMCs arising as the composition of many smaller I/O-IMCs this means that we can analyse such I/O-IMCs without computing the graph representation of the composed I/O-IMC, whose size may grow exponentially in the number of component I/O-IMCs. However, in order to apply such analysis techniques it is necessary to know beforehand *which* composed I/O-IMCs are in fact deterministic. In Chapter 8 we will discuss structural conditions on the graph level that ensure determinism, and can be computed in an efficient way.

1.3.3 Expressiveness

Finally, it is important to ensure I/O-IMCs are expressive enough to model interesting complex dependable systems. In Chapter 9 we will present the use of I/O-IMCs as an underlying semantics for the high-level modelling language ARCADE. The latter has been designed as a modelling language to describe complex dependable systems. We will also apply the theory developed in Chapter 8 to provide an efficient way of determining which ARCADE model correspond to deterministic I/O-IMCs. All in all, we demonstrate that I/O-IMCs can be used to

- model complex dependable systems by modelling their components,
- give an I/O-IMC semantics to such a system by taking the composition of the I/O-IMC semantics of its components,
- study parts of an ARCADE model in isolation, and
- prove dependability properties of the system by translating the I/O-IMC to a CTMDP or CTMC and applying standard analysis techniques.

1.4 Contribution and structure

This section presents the structure of the thesis body, discusses the novelty of its contribution, and relates it to previously published work the thesis builds on. Figure 1.1 displays how the chapters of the thesis build on each other.

- In Chapter 2 we discuss some preliminaries that will be useful throughout the thesis. In particular, we establish our notion of a state space, which is somewhat different from the state spaces used in the context of, e.g., process algebras. In this chapter we also review some fundamentals from the realm of probability theory which can be found in any textbook on the subject.
- Chapter 3 discusses Markov chains, in particular CTMCs, following the discussion of CTMCs by Anderson, Doob, and Freedman [1, 16, 17]. In this chapter, we give a proof for the correctness of the equivalence of *weak bisimulation* for CTMCs, that is applicable to a wider range of CTMCs than earlier proofs and uses a different

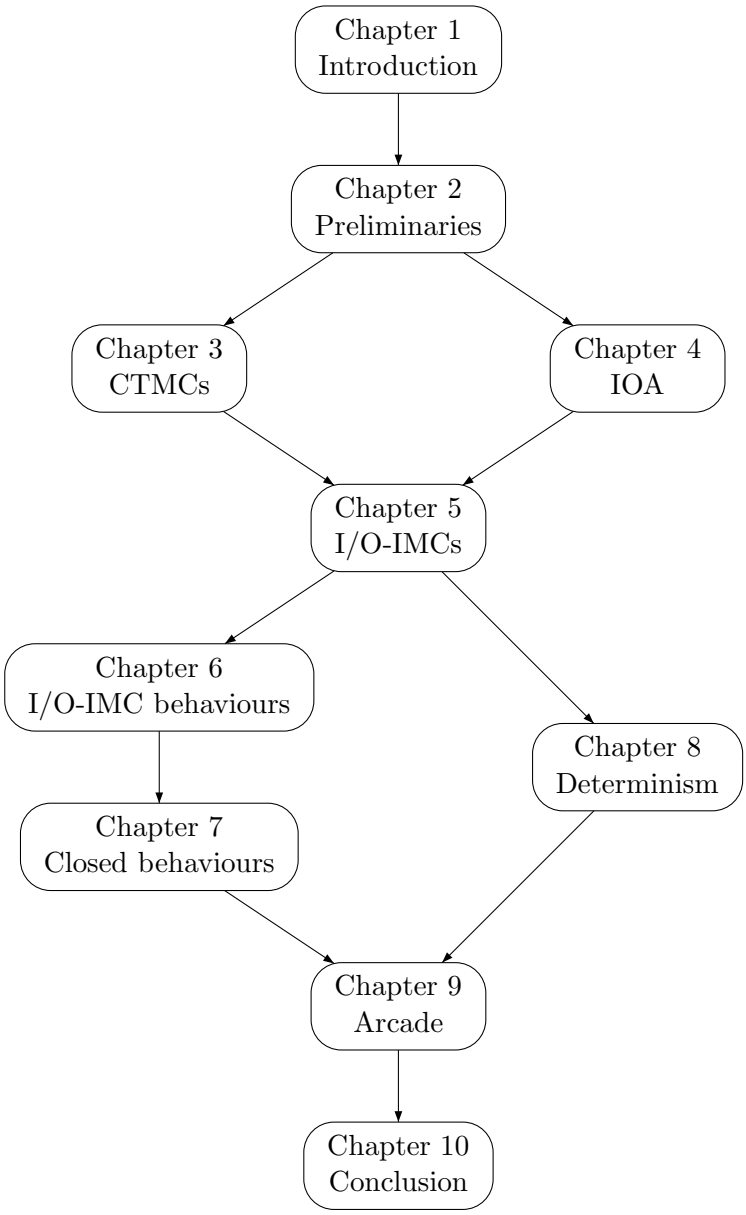


Figure 1.1: Structure of the thesis.

proof strategy which will be relevant when we revisit weak bisimulation in the context of the semantics of I/O-IMCs.

- Chapter 4 discusses a variant of IOA. It is based on the work on IOA by Lynch and Tuttle [33], but we need to make some small changes to the interpretation of IOA to facilitate combining IOA with CTMCs. We will demonstrate that these

changes do not affect the important modularity results for IOA.

- In Chapter 5 we discuss the syntax of I/O-IMCs and structural operations on it. This chapter is based on joint work with Boudali and Stoelinga [4, 5, 6, 7].
- Chapter 6 presents a modular semantics for I/O-IMCs in terms of *interactive jump processes* (a variant of jump processes, where each jump is annotated with an action-trace). The notion of interactive jump processes, their use as semantic underpinning of I/O-IMCs, and the modularity result of the I/O-IMC semantics original to the thesis and have not yet been unpublished.
- Chapter 7 focusses on the semantics of *closed* I/O-IMCs, and provides a translation to CTMDPs. This translation is based closely on a similar translation from interactive Markov chains (IMCs) to CTMDPs developed by Johr [30], which has been applied to I/O-IMCs in previous joint work [5, 6]. However, instead of using the translation to give a monolithic semantics to I/O-IMCs we establish that this translation arises naturally from the modular semantics developed in Chapter 6.
- In Chapter 8 we proceed by developing sufficient structural conditions for the determinism of a composite I/O-IMC as well as an efficient algorithm to determine whether these conditions are satisfied. This chapter represents novel and unpublished scientific insights.
- Chapter 9 introduces the high-level dependability modelling language ARCADE and its semantics in terms of I/O-IMCs. This chapter is based on previously published joint work with Boudali, Haverkort, Kuntz, and Stoelinga [3], and work by Maaß [34]

2
Preliminaries

This chapter walks through several topics which are needed for the remainder of the thesis. The first section discusses our notion of discrete states. The second section introduces standard concept from probability theory and can be easily skipped for readers with a background in probability theory. Finally, the third section briefly explains the use of Laplace transforms.

2.1 States

In essence, a state is a snapshot of the current situation of a system. For instance, if our system is a computer, a state describes the current content of registers, memory, hard disk, etc. In this section we give some mathematical structure to the collection of states we consider. Most importantly, we define a *parallel composition* operator for states. If one component of a complex system occupies a state x, and another occupies a state y, then we say that their parallel composition (i.e., the system consisting of both these components) occupies the state $x \| y$. In summary, we want to define

- an infinite set of states S_{all},
- which can be composed with a parallel operator $\|$, and
- which can be compared with an equivalence relation $=_s$ which respects $\|$.

We will now give the technical details of our interpretation of states.

Our set of all states S_{all} is induced by applying the parallel composition operator to a set S_{basic} of basic or atomic states.

Definition 1. *Let S_{basic} be a set of basic states and let $(S_{\mathsf{all}}, \|)$ be the free semigroup induced by S_{basic}. That is, S_{all} is the set of all strings of elements of S_{basic}. Since $(S_{\mathsf{all}}, \|)$ is a semigroup we have,*

- *Closure*

$$x, y \in S_{\mathsf{all}} \implies x \| y \in S_{\mathsf{all}}, \text{ and}$$

- *Associativity*

$$\forall x, y, z \in S_{\mathsf{all}} \cdot x \| (y \| z) = (x \| y) \| z.$$

Note that $(S_{\mathsf{all}}, \|)$ is the *free* semigroup, which means that two states in S_{all} are equal if and only if their equality can be derived from the associativity of the operator $\|$. We might expect the parallel composition operator to be commutative (i.e. $x \| y = y \| x$), but this would cause problems, since it is often important to know which state in a parallel composition belongs to which component.

We further assume that, to an outside observer, certain states are indistinguishable. In other words, the states are *partially observable*. Consider, for example, a computer. We can observe the computer's display and the sounds it makes, however, the exact state of the computer may include the contents of the memory and registers. We may not be able to distinguish two computers which display the same image, although the contents of, e.g., their memory may be different.

We assume there exists a congruence relation $=_s$ on $(S_{\mathsf{all}}, \|)$ such that $\|$ is commutative and transitive with respect to $=_s$ and each equivalence class of $=_s$ is infinitely large. This relation $=_s$ describes the observable part of a state. If two states are equivalent according to $=_s$ then we cannot observe a difference between them, although the states may be different according to the syntactic equivalence relation $=$. In this case, we say that such states are *observably equivalent*. Note that composite states may be observably equivalent to basic states.

For technical reasons, we will consider in this thesis only subsets of S_{all} which are of a lower dimension than S_{all} and the equivalence classes of S_{all} with respect to $=_s$. That is, we only consider sets S such that $S_{\mathsf{all}} \setminus S$ is infinitely large and the set $[x]_{=_s} \setminus S$ is also infinitely large for any equivalence class of S_{all} with respect to $=_s$. The consequence is that, for any two such subsets S_1 and S_2 we have that we can always find a set S_2' which is disjoint from S_1, but isomorphic to S_2 up to $=_s$.

It will be useful to lift the equivalence relation $=_s$ to sets of states. We interpret a set of states as a *choice* between these states. Two sets of states are then equivalent, if no matter which two representative states we choose they are always equivalent. That is, two sets $C, D \subset S_{\mathsf{all}}$ are equivalent, written – by abuse of notation – $C =_s D$, if for all pairs $x \in C$, $y \in D$ we have $x =_s y$.

Note that neither the equivalence relation $=_s$ or the equivalence $=$ make any statements about the dynamics of processes that take values in some subset of S_{all}. This means that different processes may occupy the same state at the same time, but still behave differently afterwards.

To give some insight into our reasons to assume such an equivalence relation $=_s$, we now give several examples of how it may be realized.

1. We might associate with each basic state a *reward* or *cost*. That is, we have a function $f_b : S_{\mathsf{basic}} \to \mathbb{R}$ which assigns a reward to each basic state. Now we can use f_b to induce a reward function on all states $f : S_{\mathsf{all}} \to \mathbb{R}$ by applying some

commutative and transitive operator on the reals to define the reward of composite states. For instance, we could define f to be the function induced by the axioms

$$f(x) = f_b(x), \quad x \in S_{\mathsf{basic}}$$

and

$$f(x \| y) = f(x) + f(y), \quad x, y \in S_{\mathsf{all}}.$$

Now, we can define $=_s$ as equality of rewards, i.e.,

$$x =_s y \Leftrightarrow f(x) = f(y).$$

Since addition is commutative and transitive, we have that the parallel composition operator $\|$ is commutative and transitive with respect to $=_s$. Other possible operators are multiplication, maximum, minimum, etc.

2. Assume there exists a countable set L of state-labels and a function $f_b : S_{\mathsf{basic}} \to 2^L$ which assigns a subset of state-labels to each basic state. These state-labels then describe the observable part of the state. Again we can define the state-labels of composite states by using some commutative and associative set-operator such as union or intersection. The equivalence relation $=_s$ is then defined simply as equivalence of the sets of state-labels of states.

2.2 Probability theory

In this section we describe several topics from basic probability theory. This section can be skipped for those familiar with probability theory.

2.2.1 Stochastic experiments

A stochastic experiment is an experiment whose outcome is completely determined by chance. Examples include rolling dice, flipping coins, buying lottery tickets, etc. Mathematically, we describe a stochastic experiment as a *probability space*.

Definition 2 (Probability space)**.** *A stochastic experiment can be described by a* probability space*, which is a triple* $(\Omega, \mathcal{F}, P)$*, where*

- Ω *is the set of all possible outcomes of the experiment, called the* sample space*,*
- $\mathcal{F}$ *is a set of events, called the* σ-algebra*, where each event is a subset of* Ω*, and*
  - $\mathcal{F}$ *contains* Ω *and the empty set* $\emptyset$*,*
  - $\mathcal{F}$ *is closed under complement, we have*

$$A \in \mathcal{F} \implies \Omega \setminus A \in \mathcal{F},$$

and,

  - *$\mathcal{F}$ is closed under countable union, for countably many events $\{A_i \mid i \in \mathbb{N}\}$, we have*
$$\forall i \in \mathbb{N} \cdot A_i \in \mathcal{F} \implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}.$$

- *P is a probability measure function from $\mathcal{F}$ to $[0,1]$, where*
  - *P is countably additive, for countably many* pairwise disjoint *events $\{A_i \in \mathbb{N}\}$ we have,*
$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i),$$
  and
  - *P assigns one to the set of all outcomes,*
$$P(\Omega) = 1.$$

The set Ω is the set of all possible outcomes of the experiment. Performing the experiment (e.g., rolling a die or flipping a coin) means picking an outcome ω out of Ω. The function P is used to attach probabilities to the outcomes. If Ω is finite, then we can define P such that it simply assigns a probability to each outcome and we can pick the simple σ-algebra $\mathcal{F} = \mathcal{P}(\Omega)$. An event is *measurable* if it is in $\mathcal{F}$.

Example 1. *Let's roll a fair six-sided die. We choose as the set of possible outcomes the number of pips the die shows,*
$$\Omega = \{1,2,3,4,5,6\}.$$
We use the standard σ-algebra for finite sample spaces $\mathcal{F} = \mathcal{P}(\Omega)$, i.e., $\mathcal{F}$ contains all subsets of Ω. The probability function P uniformly randomly picks an outcome, i.e.,
$$P(\{\omega\}) = 1/6, \qquad \omega \in \Omega.$$
For finite sample spaces, we can derive all other event-probabilities from the probabilities of the individual outcomes. For instance, the probability of throwing an odd number, which is described by the event $\{1,3,5\}$ can be calculated using the countable additivity of P,
$$P(\{1,3,5\}) = P(\{1\} \cup \{3\} \cup \{5\}) = P(\{1\}) + P(\{3\}) + P(\{5\}) = 1/2.$$
Note that we have some freedom in choosing the set of possible outcomes. If we throw our die onto a one meter by one meter table, we could also note the position of the die after rolling. The set of all outcomes would then be $\{1,2,3,4,5,6\} \times [0,1] \times [0,1]$ and would be uncountable! Still, we can choose the same simple σ-algebra as before, namely
$$\mathcal{F} = \{\{(i,x,y) \mid i \in v, x \in [0,1], y \in [0,1]\} \mid v \subset \{1,2,3,4,5,6\}\}.$$
For each subset v of the original sample space, we consider the event "the number of pips shown is in v and the die is located anywhere on the table". This cleverly chosen σ-algebra allows us to use our simple probability function P, despite the uncountable nature of the sample space. In general, we cannot always perform such a construction.

From a single stochastic experiment we may want to derive different properties. For instance, if we role two dice, we may be interested in the sum of pips, difference of pips, or even the product of the pips. We describe such properties as *random variables*.

Definition 3 (Discrete random variable)**.** *Given a probability space $(\Omega, \mathcal{F}, P)$ and a countable state space S, a random variable X is a function from sample space Ω to state space S, such that the inverse of X, written X^{-1} is measurable, i.e. for each $x \in S$ we have that the set,*

$$\{\omega \mid X(\omega) = x\}$$

is measurable. The probability function P then describes the probability that the random variable X takes on a particular value $x \in S$. We write $\Pr(X = x)$ for this probability and find

$$\Pr(X = x) = P(\{\omega \mid X(\omega) = x\}).$$

Example 2. *Let us roll two fair six-sided dice. We are interested in the total number of pips showing on the two dice. First we define the probability space of this experiment. We have outcomes $\Omega = \{1,2,3,4,5,6\} \times \{1,2,3,4,5,6\}$, the standard σ-algebra for finite sample spaces consisting of all subsets of Ω, and for each outcome $\langle x, y\rangle \in \Omega$ we have $P(\{\langle x, y\rangle\}) = (1/6)^2 = 1/36$. Note that we can construct this probability space from two copies of the probability space in Example 1.*

Now consider the random variable X which maps outcomes to natural numbers in the following way,

$$X(\langle x, y\rangle) = x + y,$$

for all $\langle x, y\rangle \in \Omega$. We can compute probabilities for the values of X. For instance, the probability to roll 5 pips is the probability of the event $A = \{\omega \mid \omega \in \Omega, X(\omega) = 5\}$. We have

$$\Pr(X = 5) = P(\{\langle 1,4\rangle\}) + P(\{\langle 2,3\rangle\}) + P(\{\langle 3,2\rangle\}) + P(\{\langle 4,1\rangle\}) = 4/36.$$

For uncountable state spaces, such as $\mathbb{R}_{\geq 0}$, we can define *continuous random variables* in a similar way as discrete random variables. However, the requirement of measurability is slightly more involved in the continuous case. As a final remark on random variables and probability spaces, we note that we usually have a single probability space on which all random variables are defined. If necessary, probability spaces of different experiments can easily be combined. We have seen an example of this construction in Example 2 where the probability spaces of two die-experiments where combined.

2.2.2 Basic laws of probability

We now give some often used definitions and useful laws of probability. We fix a probability space $(\Omega, \mathcal{F}, P)$ and let A, B, C be events in $\mathcal{F}$. We then use the following notations

$$\begin{aligned} \Pr(A) &\equiv P(A), \\ \Pr(A \vee B) &\equiv P(A \cup B), \\ \Pr(A \wedge B) &\equiv P(A \cap B), \text{ and} \\ \Pr(\neg A) &\equiv P(\Omega \setminus A). \end{aligned}$$

By applying results from set theory to the events we find the following results

$$\begin{aligned}\Pr(A \vee B) &= \Pr(A) + \Pr(B) - \Pr(A \wedge B),\\ \Pr(A \wedge B) &= \Pr(A \vee B) - \Pr(A \wedge \neg B) - \Pr(B \wedge \neg A), \text{ and}\\ \Pr(\neg A) &= 1 - \Pr(A).\end{aligned}$$

We complete our discussion of basic probability theory by introducing the notion of dependency between events and conditional events.

Definition 4 (Independence of events)**.** *Given a probability space* $(\Omega, \mathcal{F}, P)$*, we say two events* A *and* B *in* $\mathcal{F}$ *are independent if,*

$$Pr(A \wedge B) = \Pr(A)\Pr(B).$$

Definition 5 (Conditional probabilities)**.** *Given a probability space* $(\Omega, \mathcal{F}, P)$ *and two events* A *and* B *in* $\mathcal{F}$*, such that* $\Pr(B) > 0$*, the probability of* A under the condition B*, written* $\Pr(A \mid B)$ *is defined as the probability of* A *and* B *divided by the probability of* B

$$\Pr(A \mid B) = \frac{\Pr(A \wedge B)}{\Pr(B)}.$$

For independent events A and B we have

$$\Pr(A \mid B) = \frac{\Pr(A \wedge B)}{\Pr(B)} = \frac{\Pr(A)\Pr(B)}{\Pr(B)} = \Pr(A).$$

We find the following laws concerning conditional probabilities. Let $\{C_i \mid 1 \leq i \leq n\}$ be a countable set of events such that $\bigcup_{i=1}^{n} C_i = \Omega$. We then have

$$\begin{aligned}\Pr(A \wedge B) &= \Pr(A|B)\Pr(B),\\ \Pr(A) &= \sum_{i=1}^{n} \Pr(A \wedge C_i), \text{ and}\\ \Pr(A) &= \sum_{i=1}^{n} \Pr(A|C_i)\Pr(C_i).\end{aligned}$$

The last two laws are two different descriptions of the *law of total probability.*

As a final comment, it is important to note that any event B, such that $\Pr(B) > 0$ induces a new *conditional* probability space with probability function P_B such that, for any event $A \subset \Omega$, we have

$$P_B(A) = \Pr(A \mid B).$$

We can easily show that the function P_B is indeed a probability function.

2.2.3 Stochastic Processes

In the previous section we have seen that we can describe a single stochastic experiment using a probability space and we can describe interesting properties of a stochastic experiment as random variables. A stochastic process describes a *series* of stochastic experiments. Given a set of time-points T a stochastic process defines a random variable X for each time-point $t \in T$.

Definition 6 (Stochastic process)**.** *Given a* time-domain T*, a* stochastic process *is a family of random variables* $\{X^{(t)} \mid t \in T\}$ *defined over the same probability space* $(\Omega, \mathcal{F}, P)$ *and taking values in a set* S*, called the* state space *of the process.*

Example 3. *Let's roll two fair six-sided dice repeatedly. Let* $X^{(i)}$*, for any* $i \in \mathbb{N}$ *be the random variable that describes the sum of the pips showing after the* i*-th dice roll (as in Example 2). We first define the probability space. Each outcome is an infinitely long series of pairs from* $\{1,2,3,4,5,6\} \times \{1,2,3,4,5,6\}$*. We say the sample space* Ω *is the set of all functions from* $\mathbb{N}$ *to* $\{1,2,3,4,5,6\} \times \{1,2,3,4,5,6\}$*. Defining the* σ*-algebra and probability function is a bit more complicated as the usual strategy of defining a probability for each outcome fails. For instance, if we assign to the outcome* $\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots$ *probability* $\prod_{i=1}^{\infty} \bar{P}(\{\langle x_i, y_i\rangle\})$*, where* $\bar{P}$ *is the probability function from Example 2, then we would assign probability zero to each outcome!*

Instead, we use events that describe finitely many dice-rolls. For instance, $\mathcal{F}$ *contains the event "third dice-roll is* $\langle 3,5\rangle$ *and fifth dice-roll is* $\langle 6,4\rangle$*". This event* A *contains all the outcomes which match the description.*

$$A = \{\omega \mid \omega \in \Omega, \omega(3) = \langle 3,5\rangle, \omega(5) = \langle 6,4\rangle\}.$$

For this event we define the probability

$$P(A) = \bar{P}(\{\langle 3,5\rangle\}) \cdot \bar{P}(\{\langle 6,4\rangle\}) = (1/36)^2.$$

In general we find for an n*-dimensional event* A*, which describes the state of the process at time-points* $t_1, \ldots, t_n$*, such that the* t_i*-th dice-roll is* $\langle x_i, y_i\rangle$ *the probability*

$$P(A) = \prod_{i=1}^{n} \bar{P}(\{\langle x_i, y_i\rangle\}) = (1/36)^n.$$

The random variables $X^{(t)}$ are defined over the same probability space, so given an outcome $\omega \in \Omega$, we find for each time-point t a value $x \in S$ such that $X^{(t)}(\omega) = x$. This series of values is called a *trajectory* of the stochastic process.

Definition 7 (Trajectory)**.** *Given a stochastic process* X *and an outcome* $\omega \in \Omega$*, the* trajectory *described by* ω *is a function* $f_\omega : T \to S$ *such that*

$$f_\omega(t) = X^{(t)}(\omega).$$

Abusing the notation, we write $\omega(t)$ *for* $f_\omega(t)$*. This also matches the usual construction of the sample space* Ω *as a set of functions from* T *to* S*.*

A trajectory can be understood to be a single "run" of the stochastic process. If the time-domain is infinite, then single trajectories are usually not included as events and therefore do not have a probability. For instance, for Example 3 the probability of throwing two "ones" infinitely often is not measurable, although given the definition of P we would intuitively say the probability is zero.

We have so far left the nature of the time-domain T open. Instead of going into detail we note that there are two common choices for T: the set of all natural numbers $\mathbb{N}$, in which case T is countable and X is called a *discrete-time* stochastic process, or the set of all positive real numbers $\mathbb{R}_{\geq 0}$, in which case T is uncountable and X is called a *continuous-time* stochastic process.

Example 4. *Consider a single molecule of a radio-active material with decay-constant λ. We can model the decay of this molecule as a stochastic experiment whose outcome is a time-point $t \in \mathbb{R}_{\geq 0}$ which denotes the time of decay. We then have $\Omega = \mathbb{R}_{\geq 0}$. We consider the σ-algebra induced by the events $\{A_t \mid t \in \mathbb{R}_{\geq 0}\}$, where*

$$A_t = \{\omega \mid \omega \leq t\}.$$

From physics it is known that we find probabilities

$$P(A_t) = 1 - e^{-\lambda t}.$$

Consider the stochastic process $\left\{X^{(t)}\right\}_{t \in \mathbb{R}_{\geq 0}}$, which records the number of molecules at every point in time. That is, for an outcome $\omega \in \Omega$ we have

$$X^{(t)}(\omega) = \begin{cases} 1, & \text{if } t < \omega, \\ 0, & \text{if } t \geq \omega. \end{cases}$$

If the trajectories of a stochastic process are piecewise constant (such as the trajectories of the stochastic process in Example 4), then the stochastic process is called a *jump process*.

Definition 8. *Given a state space S and a continuous time-domain T, a stochastic process $\{X^{(t)} \mid t \in T\}$ which takes values in S is a* jump process *if its trajectories are piece-wise constant. The jump process X is called* stable *if for any state $x \in S$ and any time-point $t \in T$ we have*

$$\lim_{h \downarrow 0} \Pr(X^{(t+h)} = x \mid X^{(t)} = x) = 1$$

In fact, the process described in Example 4 is one example a stable jump process. We will now give another example of a jump process.

Example 5. *Consider a queue in a convenience store. At any time zero or more customers may be waiting in the queue depending on when they join the queue and how fast customers pay for their purchases at the register. We will consider the stochastic process $\{X^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$ which describes the number of customers in the queue at any*

given time-point. We will choose $\Omega = \mathbb{N}_0$ *as the sample space of all* $X^{(t)}$ *and we will use the standard sigma algebra consisting of all subsets of* Ω.

Let's say that our first customer joins the queue at time-point Y_1 *and the second customer joins the queue* Y_2 *time-points later and so forth. The time it takes for the customers to pay at the register is given by* Z_1, Z_2, *etc. Now assume that the arrival times are independent random variables that are all uniform distributed between* 2 *and* 6 *time-units, i.e. for all* $i \in \mathbb{N}$ *we have*

$$\Pr(Y_i \leq t) = \frac{\max(\min(t, 6), 2) - 2}{4}.$$

Furthermore we have that every customer spends either 3 *or* 5 *time-units at the cash register (depending on how they pay). We will assume that each customer has a 50% chance of spending either* 3 *or* 5 *time-units at the cash register:*

$$\Pr(Z_i = n) = \begin{cases} 1/2, & \text{if } n = 3 \text{ or } n = 5, \\ 0, & \text{otherwise.} \end{cases}$$

Figure 2.1 gives an example of a trajectory of X. *It should be clear that* X *is a jump process since its trajectories are indeed piece-wise constant. This example illustrates that the distribution of the times between jumps of a jump-process does not matter as long as it is not zero (i.e., is zero with probability zero).*

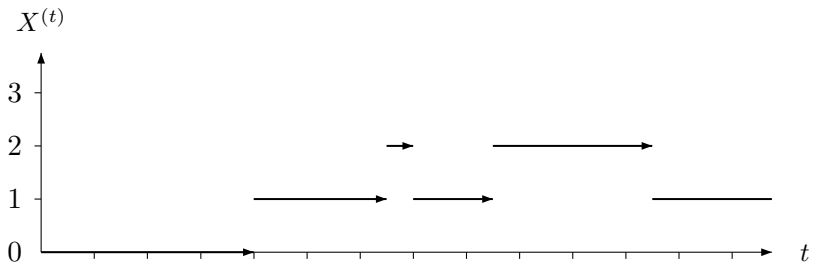


Figure 2.1: Example of a trajectory for the jump process X which describes the number of customers in a queue. The first customer arrives after 4 times units and takes 3 time-units to pay (leaving the queue at $t = 7$). The second customer arrives 2.5 time-units after the first one (at $t = 6.5$) and leaves 5 time-units later (at $t = 11.5$). A third customer arrives 2 time-units after the second and leaves after 5.5 time-units (not shown).

2.3 Laplace transform

The Laplace transform is an alternative way to represent functions. In certain situations it is easier to work with the Laplace transform of a function than with the function itself. We now briefly discuss Laplace transforms without going into detail, since we will use Laplace transforms in Chapter 3.

Definition 9. *Given a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$, its Laplace transform is a function $F : \mathbb{R}_{\geq 0} \to \mathbb{R}$, with*

$$F(s) = \int_0^\infty e^{-st} f(t) dt,$$

for all $s \in \mathbb{R}_{\geq 0}$.

We now give a list of Laplace transforms that we will use in Section 3.2. Below, c is a constant and g and h are functions with respective Laplace transforms G and H.

Function	Laplace transform
$f(t) = c$	$F(s) = \frac{c}{s}$
$f(t) = 1 - e^{-ct}$	$F(s) = \frac{c}{s(s+c)}$
$f(t) = cg(t)$	$F(s) = cG(s)$
$f(t) = g(t) + h(t)$	$F(s) = G(s) + H(s)$
$f(t) = \frac{d}{dt} g(t)$	$F(s) = sG(s) - g(0)$.

3
Continuous-time Markov chains

As noted in Chapter 1, Markov chains are a versatile and widely-used way of modeling a variety of stochastic phenomena. In this chapter we will discuss continuous-time Markov chains (CTMCs) in more detail.

Contribution. The first section of this chapter reiterates results on Markov chains and is adapted mainly from Anderson [1]. We revisit several key proofs for CTMCs from Anderson to set the stage for several similar proofs that we will need for our compositional models in Chapters 6 and 7. A genuine contribution is our study of bisimulation for countable infinite-state continuous-time Markov chains. We prove that bisimulation preserves transient probabilities for infinite-state Markov chains provided the equivalence classes are all regular. For irregular Markov chains (that do not have a unique solution), we show that bisimulation preserves the minimal solution to their forward and backward equations (see (3.11) respectively (3.9)), if all equivalence classes of the bisimulation are regular. Regularity of an equivalence class means that if we construct a Markov chain from such a class it will have a unique transient solution for each time-point.

3.1 Continuous-time Markov chains

Definition 10. *Given a countable state space S, a continuous-time Markov process (or* chain*) is a stochastic process $\{X^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$, such that we find for any states $y, x_1, \ldots, x_n \in S$ and any series of time-points $t > t_n > \ldots > t_1 \in \mathbb{R}_{\geq 0}$ that:*

$$\Pr(X^{(t)} = y \mid X^{(t_n)} = x_n, \ldots, X^{(t_1)} = x_1) = \Pr(X^{(t)} = y \mid X^{(t_n)} = x_n). \tag{3.1}$$

We do not yet define a probability space $(\Omega, \mathcal{F}, P)$ for this continuous-time Markov chain. There are different ways of constructing such a probability space [17]. However,

we will see that it is rarely necessary to work with the probability space of a Markov chain directly. Instead, we will make use of certain fundamental probabilities that can be derived from (3.1).

The property (3.1) is called the Markov property. For a stochastic process that fulfils the Markov property we have that the probability of reaching a state y at time t after occupying a state x_n at time t_n does not depend on the value of the process before time t_n. This means a Markov chain is *memoryless* and we need only know the current state a Markov chain occupies to determine its future behaviour. We can also say that, considering the probability space under the condition $\{X^{(t_n)} = x_n\}$, the event $\{X^{(t)} = y\}$ (which describes the future w.r.t. t_n) is independent of the event $\{X^{(t_{n-1})} = x_{n-1}, \ldots, X^{(t_1)} = x_1\}$ (which lies in the past w.r.t. t_n). The Markov property (3.1) then follows.

Example 6. *The Markov property plays a critical role in Markov process theory. Here are a few examples of how the Markov property can (and cannot) be used. Below x, y, z are states in S and t_1, t_2, t_3 are time-points in $\mathbb{R}_{\geq 0}$ such that $t_1 < t_2 < t_3$.*

First, the Markov property can be applied to "uncountable" conditional probabilities. For instance, let's look at the probability to be in state y at time-point t_3 under the condition that the Markov chain occupied state x from time-point t_1 to time-point t_2. We can then apply the Markov property to find that this probability only depends on the fact that X was in x at time t_2, not how long it occupied this state.

$$\Pr(X^{(t_3)} = y \mid X^{(t)} = x, t_1 \leq t \leq t_2) = \Pr(X^{(t_3)} = y \mid X^{(t_2)} = x).$$

It is easy to show, using the laws of probability, that conditional probabilities are also independent of the fact that X occupied some subset D of the state space S at an earlier point in time.

$$\Pr(X^{(t_3)} = y \mid X^{(t_2)} = x \wedge X^{(t_1)} \in D) = \Pr(X^{(t_3)} = y \mid X^{(t_2)} = x).$$

However, the reverse does not *hold. The probability to be in a state y at time t_3, given that X occupied a state in subset D at time t_2 and a state x at time t_1 is* not *independent of the fact that X was in x at t_1. It may then be the case that,*

$$\Pr(X^{(t_3)} = y \mid X^{(t_2)} \in D \wedge X^{(t_1)} = x) \neq \Pr(X^{(t_3)} = y \mid X^{(t_2)} \in D).$$

The reason we can not apply the Markov property is the following. The fact that the Markov chain occupies x at time t_1 influences which *state in D is occupied at time t_2 and the states in D may have different probabilities to reach y at time t_3. We can show*

this with a simple calculation,

$$\begin{aligned}\Pr(X^{(t_3)}=y \mid X^{(t_2)}\in D \wedge X^{(t_1)}=x) &= \frac{\Pr(X^{(t_3)}=y \wedge X^{(t_2)}\in D \wedge X^{(t_1)}=x)}{\Pr(X^{(t_2)} \in D \wedge X^{(t_1)}=x)} \\ &= \frac{\sum_{z\in D}\Pr(X^{(t_3)}=y \wedge X^{(t_2)}=z \wedge X^{(t_1)}=x)}{\Pr(X^{(t_2)} \in D \wedge X^{(t_1)}=x)} \\ &= \frac{\sum_{z\in D}\Pr(X^{(t_3)}=y \mid X^{(t_2)}=z \wedge X^{(t_1)}=x)\Pr(X^{(t_2)}=z \wedge X^{(t_1)}=x)}{\Pr(X^{(t_2)} \in D \wedge X^{(t_1)}=x)} \\ &= \sum_{z\in D}\Pr(X^{(t_3)}=y \mid X^{(t_2)}=z \wedge X^{(t_1)}=x)\cdot \\ &\qquad \Pr(X^{(t_2)}=z \mid X^{(t_2)} \in D \wedge X^{(t_1)}=x)\end{aligned}$$

We can apply the Markov property to find the above equals,

$$\sum_{z\in D}\Pr(X^{(t_3)}=y \mid X^{(t_2)}=z)\Pr(X^{(t_2)}=z \mid X^{(t_2)} \in D \wedge X^{(t_1)}=x)$$

Now we find, that the above equals $\Pr(X^{(t_3)} = y \mid X^{(t_2)} \in D)$*, if the probabilities* $\Pr(X^{(t_3)} = y \mid X^{(t_2)} = z)$ *are equal for all states* $z \in D$*. In general, this is not the case.*

Given a random variable J *which takes values in* $\mathbb{R}_{\geq 0}$ *and which depends* only *on values of* X *at times smaller or equal to* J*, we say* J *is a* stopping-time *of* X*. For stopping-times we can also apply the Markov property. I.e., we have that the Markov chain after* J *is independent of the Markov chain before* J*. Let* t_1 *be smaller than* J *and* t_3 *greater than* J*, then*

$$\Pr(X^{(t_3)} = y \mid X^{(J)} = x \wedge X^{(t_1)} = z) = \Pr(X^{(t_3)} = y \mid X^{(J)} = x).$$

The above is called the strong Markov property*. Usually, the condition that* $t_3 > J > t_1$ *follows from the definition of the random variable* J*.*

Finally we note that the Markov property must be applied to the largest *time-point in the condition. I.e., we may find that*

$$\Pr(X^{(t_3)} = y \mid X^{(t_2)} = z \wedge X^{(t_1)} = x) \neq \Pr(X^{(t_3)} = y \mid X^{(t_1)} = x).$$

We have seen that the future behaviour of a Markov chain does not depend on the past behaviour. However, it may depend on the current time. If, instead, we have for any two states $x, y \in S$ and time-points $t_1, t_2, t_3 \in \mathbb{R}_{\geq 0}$ that the probability to reach y from x does *not* depend on the current time, i.e.,

$$\Pr(X^{(t_2)} = y \mid X^{(t_1)} = x) = \Pr(X^{(t_2+t_3)} = y \mid X^{(t_1+t_3)} = x) \tag{3.2}$$

then the Markov chain is called *time-homogeneous*. For time-homogeneous Markov chains we have that their future behaviour does not depend on the current time, only on the current state of the Markov chain. In the following, we consider a time-homogeneous continuous-time Markov chain X with countable state space S.

We consider only Markov chains which are jump processes. This means that for all trajectories we have that they are piecewise-constant and right-continuous. In essence this means that a run of a Markov chain behaves as follows. The Markov chain starts in a particular state x, stays in x for a non-zero period of time and then (possibly) jumps to a different state y where it again stays for a non-zero period of time before possibly jumping to another state, and so forth.

3.1.1 Describing a Markov chain

Before we dive into Markov chain theory, we quickly give an overview of the different ways to describe Markov chains. Since a Markov chain is a family of random variables on the same probability space, the first thing we need is to define the probability space $(\Omega, \mathcal{F}, \bar{P})$. Even if we restrict to piece-wise constant trajectories there are still uncountably many possible trajectories in Ω, each of which is a function from the uncountably large time-domain $\mathbb{R}_{\geq 0}$ to the countable state space S (see Figure 3.1). The σ-algebra $\mathcal{F}$ could be generated by organising the trajectories into events of the form $\{X^{(t_1)} = x_1 \wedge \ldots \wedge X^{(t_n)} = x_n\}$, for finite series of time-points and states. As the time-points are taken from $\mathbb{R}_{\geq 0}$, we have that $\mathcal{F}$ is uncountably large. Finally, the probability function $\bar{P}$ assigns a probability to each event. Given such a probability space, it is easy to define the random variables: $X^{(t)}(\omega) \equiv \omega(t)$ for each $t \in \mathbb{R}_{\geq 0}$ and $\omega \in \Omega$.

From the above discussion it should be clear that it is very challenging to define a Markov chain directly through its probability space. This is why Markov chains are often described in terms of their properties. First, we have that each time-homogeneous Markov chain has a *transition function* P which describes the probability to go from one state to another in a specific time-period,

$$P_{x,y}(t) \equiv \Pr(X^{(t)} = y \mid X^{(0)} = x),$$

where x and y are states in S and t is a time-point in $\mathbb{R}_{\geq 0}$. We can see that the transition function is much less complicated than the probability space of a Markov chain, although it is still uncountably large (see Figure 3.1). We will discuss transition functions in Subsection 3.1.2, where we will derive recursive definitions for P which can be used in certain situations. In general, a transition function does not uniquely define a continuous-time Markov chain. However, if the transition function has finite derivatives at time-point zero, we say it is *standard* and we then have that it, in a sense, "uniquely defines" a Markov chain. For this thesis we are not interested in Markov chains with non-standard transition functions, as they do not appear often in practical applications. A Markov chain with a standard transition function is called *stable*. For more details on unstable Markov chains we refer to Anderson [1].

The final Markov chain representation we discuss is the most widely used in practical applications, the infinitesimal generator Q. The infinitesimal generator is a $|S|$ by $|S|$ real-valued matrix which contains, at entry $q_{x,y}$ where x and y are states in S, the derivative at time zero of the transition function P,

$$q_{x,y} \equiv \left.\frac{d}{dt} P_{x,y}(t)\right|_{t=0}.$$

Since S is countable, Q is also countably large. Moreover, if S is finite, Q is finite and can be specified directly (see Figure 3.1). In Subsection 3.1.3 we will see that every transition function has an infinitesimal generator, and then every Markov chain has an infinitesimal generator. Conversely, an infinitesimal generator matrix uniquely defines the "finite-jump" probabilities of a Markov chain. That is, from Q we can derive probabilities

$$P_{x,y}^{(n)}(t) \equiv \Pr(X^{(t)} = y \wedge \text{“}X \text{ makes at most } n \text{ jumps in } [0,t]\text{”} \mid X^{(0)} = x)$$

for states x, y, a time-point t, and a natural number n. We will discuss this derivation in Subsection 3.1.4. Unfortunately, there are so-called *irregular* Markov chains that may perform infinitely many jumps in a finite amount of time. However, we will see that if the infinitesimal generator Q of a Markov chain is regular, then the Markov chain is also regular and can only perform finitely many steps in a finite amount of time. In this case Q uniquely defines P. We will discuss sufficient and sometimes necessary conditions for the regularity of Q in Subsection 3.1.5.

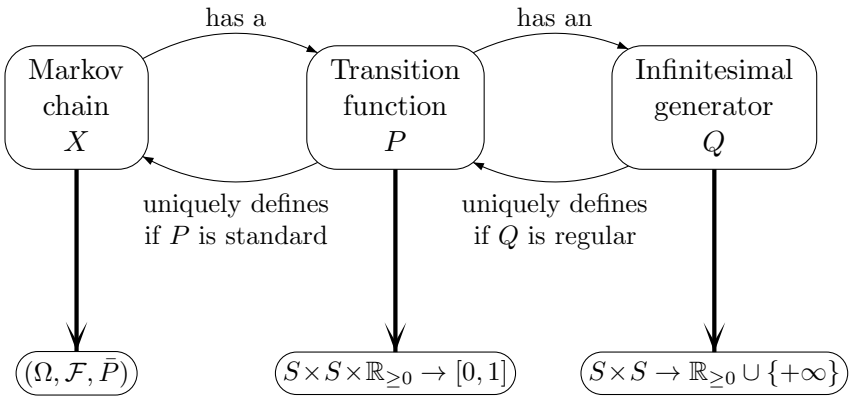


Figure 3.1: Overview of different ways of describing a Markov chain and the mathematical objects associated with each description.

3.1.2 Transition probabilities

We now consider the *transition probabilities* of a homogeneous continuous-time Markov chain X with state space S, which are described by the transition function P.

Definition 11. *For states x, y in S and a time-point $t \in \mathbb{R}_{\geq 0}$, the transition function P describes the probability that the Markov chain X occupies state y at time t under the condition that X occupies state x at time* 0

$$P_{x,y}(t) \equiv \Pr(X^{(t)} = y \mid X^{(0)} = x).$$

We can express the probability that X occupies states $x_1, \ldots, x_n$ at time-points $t_1 < \ldots < t_n$ under the condition that X occupies state x_0 at time $t_0 < t_1$ in terms of the transition function of X. We have

$$\begin{aligned}&\Pr(X^{(t_n)} = x_n \wedge X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \mid X^{(t_0)} = x_0)\\ =&\Pr(X^{(t_n)} = x_n \mid X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \wedge X^{(t_0)} = x_0)\\ &\cdot \Pr(X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \mid X^{(t_0)} = x_0).\end{aligned}$$

Applying the Markov property we find

$$\begin{aligned}&\Pr(X^{(t_n)} = x_n \mid X^{(t_{n-1})} = x_{n-1})\\ &\cdot \Pr(X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \mid X^{(t_0)} = x_0).\end{aligned}$$

Because X is time-homogeneous this is equivalent to

$$\begin{aligned}&\Pr(X^{(t_n - t_{n-1})} = x_n \mid X^{(0)} = x_{n-1})\\ &\cdot \Pr(X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \mid X^{(t_0)} = x_0).\end{aligned}$$

Following this approach we arrive at

$$\begin{aligned}&\Pr(X^{(t_n)} = x_n \wedge X^{(t_{n-1})} = x_{n-1} \wedge \ldots \wedge X^{(t_1)} = x_1 \mid X^{(t_0)} = x_0) =\\ &\prod_{i=1}^{n} P_{x_{i-1},x_i}(t_i - t_{i-1}).\end{aligned}$$

Given an initial distribution α such that $P(X^{(0)} = x) = \alpha_x$ we have that the probability that X occupies states $x_n, \ldots, x_0$ at times $t_n > \ldots > t_0$ equals

$$\sum_{x \in S} \alpha_x P_{x,x_0}(t_0) \prod_{i=1}^{n} P_{x_{i-1},x_i}(t_i - t_{i-1}). \tag{3.3}$$

From the Markov property we can also derive the following for $t_1, t_2 \in \mathbb{R}_{\geq 0}$

$$\begin{aligned}\Pr(X^{(t_1+t_2)} = y \mid X^{(0)} = x) &= \sum_{z \in S} \Pr(X^{(t_1+t_2)} = y \wedge X^{(t_1)} = z \mid X^{(0)} = x)\\ &= \sum_{z \in S} \Pr(X^{(t_1+t_2)} = y \mid X^{(t_1)} = z \wedge X^{(0)} = x)\\ &\quad \cdot \Pr(X^{(t_1)} = z \mid X^{(0)} = x).\end{aligned}$$

Applying (3.1) and (3.2) we now have

$$\begin{aligned}&\Pr(X^{(t_1+t_2)} = y \mid X^{(0)} = x) =\\ &\sum_{z \in S} \Pr(X^{(t_2)} = y \mid X^{(0)} = z) \Pr(X^{(t_1)} = z \mid X^{(0)} = x).\end{aligned} \tag{3.4}$$

This equation is known as the Chapman-Kolmogorov equation. Expressed in terms of the transition function of X we have

$$P_{x,y}(t_1 + t_2) = \sum_{z \in S} P_{x,z}(t_1) P_{z,y}(t_2). \tag{3.5}$$

We have seen that using the transition function of a continuous-time Markov chain we can determine the probability that the Markov chain occupies countably many states at countably many time-points. In fact, continuous-time Markov chains are almost completely described by these transition probabilities. We also find, from basic probability theory and the Chapman-Kolmogorov equation that the transition function of a stable and time-homogeneous Markov chain has the following properties.

1. $P_{x,y}(t) \geq 0$ for all $x, y \in S$ and $t \in \mathbb{R}_{\geq 0}$ and $\sum_{y \in S} P_{x,y}(t) = 1$ for all $x \in S$ and $t \in \mathbb{R}_{\geq 0}$,
2. for all pairs of distinct states $x, y \in S$ and time-points $t \in \mathbb{R}_{\geq 0}$, we have $P_{x,y}(0) = 0$ and $P_{x,x}(0) = 1$, and
3. for all pairs of distinct states $x, y \in S$, we find that $\lim_{t \downarrow 0} P_{x,x}(t) = 1$ and $\lim_{t \downarrow 0} P_{x,y}(t) = 0$.

Any function that satisfies the first two properties stated above and the Chapman-Kolmogorov equation, (3.5), is called a *transition function* and it is called a *standard transition function* if it also satisfies the third property [1]. As noted earlier, we will consider only Markov chains with standard transition functions in this thesis. We will now study the derivative of the transition function of a Markov chain.

3.1.3 Infinitesimal transition probabilities

Given a standard transition function P on a state space S (not necessarily associated with a Markov chain), the following limit exists, for a state $x \in S$:

$$q_x = \lim_{t \downarrow 0} \frac{1 - P_{x,x}(t)}{t} \tag{3.6}$$

It is important to note that although the above limit exists, it may not be finite. If for some state x we have that q_x is finite we say that x is *stable*. For a stable state x we find, for any state $y \in S$ such that $x \neq y$ that the limit

$$q_{x,y} = \lim_{t \downarrow 0} \frac{P_{x,y}(t)}{t} \tag{3.7}$$

exists and is finite. For proofs of the above two statements we refer to Anderson [1].

Definition 12 (Stability)**.** *Given a standard transition function $P_{x,y}(t)$ on a state space S, a state $x \in S$ is stable if q_x is finite. The transition function itself is called stable if all states in S are stable. Finally, a Markov chain X with transition function P is called stable if P is stable.*

Without proof we note that given a stable standard transition function with finite derivatives, we can always construct a time-homogeneous continuous-time Markov chain X with right-continuous, piecewise constant trajectories, such that for states $x, y \in S$ and time-point $t \in \mathbb{R}_{\geq 0}$ we have

$$P(X^{(t)} = y \mid X^{(0)} = x) = P_{x,y}(t).$$

That is, for any stable and standard transition function P there exists a Markov chain whose transition probabilities are described by P.

We say that a stable transition function "uniquely defines" a Markov chain, although in fact different Markov chains (with different sample spaces for instance) may be derived from one transition function. However, all such Markov chains have identical finite-state probabilities as given by (3.3). As for the class of Markov chains that are not stable (unstable), we do not consider them here. Note that we will use the terms *stable* and *unstable* in a different context in Chapter 5.

For distinct states $x, y \in S$, the values $q_{x,y}$ are very important for the Markov chain X. First of all we have that they are the derivatives of the functions $P_{x,y}$ at time-point zero. We have

$$\frac{d}{dt}P_{x,y}(t)\bigg|_{t=0} = \lim_{h \downarrow 0} \frac{P_{x,y}(h) - P_{x,y}(0)}{h}$$

and we have that $P_{x,y}(0)$ equals zero which means

$$\frac{d}{dt}P_{x,y}(t)\bigg|_{t=0} = q_{x,y}.$$

We find a similar result for the derivative at zero of $P_{x,x}$. We have

$$\frac{d}{dt}P_{x,x}(t)\bigg|_{t=0} = \lim_{h \downarrow 0} \frac{P_{x,x}(h) - P_{x,x}(0)}{h}.$$

Now we have that $P_{x,x}(0) = 1$ and so we have

$$\frac{d}{dt}P_{x,x}(t)\bigg|_{t=0} = -q_x.$$

We define $q_{x,x} \equiv -q_x$, for $x \in S$ and summarise the above results by defining the *infinitesimal generator matrix* of a Markov chain.

Definition 13 (infinitesimal generator matrix)**.** *Given a stable and time-homogeneous Markov chain X with state space S and transition function P. The* infinitesimal generator matrix $Q \in \mathbb{R}^{|S| \times |S|}$ *is the derivative of P at time* 0*.*

$$\frac{d}{dt}P(t)\bigg|_{t=0} = Q.$$

In general the entries of Q may be infinite, but if for all states x, $q_{x,x}$ is finite, then every entry in Q is finite, and the associated transition function is stable.

Forward and backward equations. We now derive two important connections between the infinitesimal generator and the transition function of a Markov chain, called the forward and backward (Kolmogorov) equations.

Given a Markov chain X with state space S and transition function P, let x and y be distinct states in S and let $h \in \mathbb{R}_{\geq 0}$ be a time-point. Consider the Taylor-expansion around the origin of $P_{x,y}(h)$. We have

$$P_{x,y}(h) = P_{x,y}(0) + P'_{x,y}(0)h + \frac{1}{2}P''_{x,y}(0)h^2 + \frac{1}{6}P'''_{x,y}(0)h^3 \ldots,$$

where $P'_{x,y}(0)$, $P''_{x,y}(0)$, etc. denote the first, second, etc. derivatives of $P_{x,y}(h)$ at $h = 0$. Since the derivatives at $h = 0$ are independent of h we have

$$P_{x,y}(h) = P_{x,y}(0) + P'_{x,y}(0)h + o(h)$$

where $o(h)$ denotes a function f such that $f(0) = 0$ and $\lim_{h\downarrow 0} f(h)/h = 0$. We now find

$$P_{x,y}(h) = \begin{cases} q_{x,y}h + o(h) & \text{, if } x \neq y \\ 1 - q_x h + o(h) & \text{, if } x = y. \end{cases} \tag{3.8}$$

By the law of total probability we have $\sum_{y\neq x} q_{x,y}h + o(h) + 1 - q_x h + o(h) = 1$ for any h. Without proof we note that it follows that $q_x = \sum_{y\neq x} q_{x,y}$ and then $q_{x,x} = -\sum_{y\neq x} q_{x,y}$.

We now turn our attention to the derivative of P at any time-point t. For a time-interval $h > 0$ we have

$$\begin{aligned} P_{x,y}(t+h) &= \sum_{z\in S} P_{x,z}(h)P_{z,y}(t) \\ &= \sum_{z\neq x}(q_{x,z}h + o(h))P_{z,y}(t) + (1 - q_x h + o(h))P_{x,y}(t) \\ &= \sum_{z\neq x}(q_{x,z}h + o(h))P_{z,y}(t) + (q_{x,x}h + o(h))P_{x,y}(t) + P_{x,y}(t) \\ &= \sum_{z\in S}(q_{x,z}h + o(h))P_{z,y}(t) + P_{x,y}(t). \end{aligned}$$

We can use the above to compute the derivative of $P_{x,y}(t)$

$$\begin{aligned} \frac{d}{dt}P_{x,y}(t) &= \lim_{h\downarrow 0} \frac{P_{x,y}(t+h) - P_{x,y}(t)}{h} \\ &= \lim_{h\downarrow 0} \frac{\sum_{z\in S}(q_{x,z}h + o(h))P_{z,y}(t)}{h}. \end{aligned}$$

Given that $\lim_{h\downarrow 0} o(h)/h = 0$ we have

$$\frac{d}{dt}P_{x,y}(t) = \sum_{z\in S} q_{x,z}P_{z,y}(t) \tag{3.9}$$

or, written in matrix-form

$$\frac{d}{dt}P(t) = QP(t). \tag{3.10}$$

This equation is called the *backward equation* or *Kolmogorov backward equation*.

A similar result follows if we split the time-interval $t + h$ differently, i.e. if we use,

$$P_{x,y}(t+h) = \sum_{z \in S} P_{x,z}(t) P_{z,y}(h).$$

We then find

$$\frac{d}{dt}P_{x,y}(t) = \sum_{z \in S} P_{x,z}(t) q_{z,y} \tag{3.11}$$

or, written in matrix-form

$$\frac{d}{dt}P(t) = P(t)Q. \tag{3.12}$$

This equation is called the *forward equation* or *Kolmogorov forward equation*.

The forward Kolmogorov equation can be used to describe the dynamics of the transient distribution of X. For a state $x \in S$ and a time-point $t \in \mathbb{R}_{\geq 0}$, we write

$$\pi_x(t) \equiv \Pr(X^{(t)} = x).$$

The vector $\pi(t)$ is known as the *transient distribution* of X and describes the probability that X occupies a given state at time t. From the Chapman-Kolmogorov equation it then follows that

$$\pi_y(t_1 + t_2) = \sum_{x \in S} \pi_x(t_1) P_{x,y}(t_2),$$

for states $x, y \in S$ and time-points $t_1, t_2 \in \mathbb{R}_{\geq 0}$.

We now multiply the left- and right-hand sides of (3.11) from the left with the initial probability $\pi_x(0)$. We have

$$\pi_x(0)\frac{d}{dt}P_{x,y}(t) = \pi_x(0) \sum_{z \in S} P_{x,z}(t) q_{z,y}$$

and then

$$\frac{d}{dt}\pi_y(t) = \sum_{z \in S} \pi_z(t) q_{z,y}. \tag{3.13}$$

In matrix form we find

$$\frac{d}{dt}\pi(t) = \pi(t)Q. \tag{3.14}$$

The above suggest that we can compute the transient probabilities $\pi_x(t)$ of X by solving the system of differential equations (3.14). However, we will see that the forward equation (3.12) and then also (3.14) do not always have a unique solution. Before we discuss under what conditions the forward and backward equations have unique solutions, we first discuss several further properties of a Markov chain that can be derived from its infinitesimal generator matrix.

Jump-times and jump-probabilities. For a natural number n, let J_n be a random variable describing the n-th *jump-time* of X, i.e., for an outcome ω, $J_n(\omega)$ is the time at which $X(\omega)$ changes value for the n-th time. We can define J_n inductively as follows

$$J_n = \begin{cases} 0 & \text{, if } n = 0 \\ \inf\{t \mid t > J_{n-1} \wedge X^{(t)} \neq X^{(J_{n-1})}\} & \text{, otherwise.} \end{cases} \tag{3.15}$$

We see from the definition that the random variables J_n, for $n \in \mathbb{N}$, are stopping-times of X. That is, for an outcome ω the value of $J_n(\omega)$ can be derived by inspecting the values of $X^{(t)}(\omega)$ up to $J_n(\omega)$.

In particular, J_1 is the time of the first jump of the Markov chain. Note that, because of time-homogeneity we have:

$$\Pr(J_n - J_{n-1} \leq t \mid X^{(J_{n-1})} = x) = \Pr(J_1 \leq t \mid X^{(0)} = x).$$

We now investigate the distribution of J_1 under the condition that X starts in a particular stable state $x \in S$. This distribution,

$$\Pr(J_1 \leq t \mid X^{(0)} = x)$$

is also called the *residence distribution* of state x. We will first investigate the derivative $\frac{d}{dt}\Pr(J_1 > t \mid X^{(0)} = x)$. To do this we develop a "backward" equation for this distribution. For time-points $t, h \in \mathbb{R}_{\geq 0}$ such that $h > 0$ we find

$$\begin{aligned} &\Pr(J_1 > t + h \mid X^{(0)} = x) \\ &\quad = \sum_{y \in S} \Pr(J_1 > t + h \wedge X^{(h)} = y \mid X^{(0)} = x) \\ &\quad = \sum_{y \in S} \Pr(J_1 > t + h \mid X^{(h)} = y \wedge X^{(0)} = x) \Pr(X^{(h)} = y \mid X^{(0)} = x). \end{aligned}$$

Now we have for $y \neq x$, that the event $\{X^{(h)} = y \wedge X^{(0)} = x\}$ implies that at least one jump occurred in the time-interval $[0, h]$. Obviously, the first jump-time J_1 cannot be greater than $t + h$. It follows, that

$$\begin{aligned} &\Pr(J_1 > t + h \mid X^{(0)} = x) \\ &\quad = \Pr(J_1 > t + h \mid X^{(h)} = x \wedge X^{(0)} = x) \Pr(X^{(h)} = x \mid X^{(0)} = x) \\ &\quad = \Pr(J_1 > t + h \mid X^{(h)} = x) \Pr(X^{(h)} = x \mid X^{(0)} = x). \end{aligned}$$

Note that we can apply the Markov property above since the event $\{J_1 > t + h\}$ under the condition that $X^{(h)} = x$ can be interpreted as the event $\{X^{(s)} = x \mid h \leq s \leq t + h\}$. We now apply (3.8) and the homogeneity of X to find

$$\begin{aligned} &\Pr(J_1 > t + h \mid X^{(0)} = x) \\ &\quad = \Pr(J_1 > t + h \mid X^{(h)} = x)(1 - q_x h + o(h)) \\ &\quad = \Pr(J_1 > t \mid X^{(0)} = x) - \Pr(J_1 > t \mid X^{(0)} = x)(q_x h + o(h)) \end{aligned}$$

For the derivative of the conditional distribution of J_1 we now find

$$\frac{d}{dt}\Pr(J_1 > t \mid X^{(0)} = x) = \lim_{h\downarrow 0} \frac{-\Pr(J_1 > t \mid X^{(0)} = x)(q_x h + o(h))}{h}$$
$$= -q_x \Pr(J_1 > t \mid X^{(0)} = x). \tag{3.16}$$

Since x is stable we have $\Pr(J_1 > 0 \mid X^{(0)} = x) = 1$ and q_x is finite. We then find the following unique solution for the ordinary differential equation (3.16)

$$\Pr(J_1 > t \mid X^{(0)} = x) = e^{-q_x t} \tag{3.17}$$

or $P(J_1 \leq t \mid X^{(0)} = x) = 1 - e^{-q_x t}$. We say that the *residence time* of state x has a negative exponential distribution with *rate* q_x.

We have seen that the time until X leaves a particular stable state x is determined completely by q_x. For this reason we refer to the value $q_x = -q_{x,x}$ as the *exit-rate* of state x. Recall that we have $q_x = \sum_{y\neq x} q_{x,y}$. Now the question arises, if X leaves a state x at time J_1, then what is the next state of X? In particular, we wish to know the probability that X jumps to a state y on its first jump, knowing that it started in state x,

$$\Pr(X^{(J_1)} = y \mid X^{(0)} = x). \tag{3.18}$$

Note that by time-homogeneity we have, for any $n \in \mathbb{N}$:

$$\Pr(X^{(J_n)} = y \mid X^{(J_{n-1})} = x) = \Pr(X^{(J_1)} = y \mid X^{(0)} = x).$$

The probabilities (3.18) are called the *jump-probabilities* of X.

For any positive time-interval $h < J_1$ we know that if $X^{(0)} = x$ then also $X^{(J_1-h)} = x$ and furthermore $X^{(J_1)} \neq x$ then

$$\Pr(X^{(J_1)} = y \mid X^{(0)} = x)$$
$$= \Pr(X^{(J_1)} = y \mid X^{(J_1)} \neq x \wedge X^{(J_1-h)} = x \wedge X^{(0)} = x).$$

Since the trajectories of X are right-continuous we find that the above also equals

$$\lim_{h\downarrow 0} \Pr(X^{(J_1)} = y \mid X^{(J_1)} \neq x \wedge X^{(J_1-h)} = x \wedge X^{(0)} = x).$$

Recall that we can apply the Markov property for $X^{(J_1-h)}$, because $J_1 - h$ approaches J_1 and J_1 is a *stopping time* of X[1]. Applying the strong Markov property and time-homogeneity of X we find that the jump-probability from x to y equals

$$\lim_{h\downarrow 0} \Pr(X^{(h)} = y \mid X^{(h)} \neq x \wedge X^{(0)} = x).$$

[1]Note, in particular that we cannot apply the Markov property when h does not approach zero as $J_1 - h$ is in general not a stopping time.

Now we can rewrite to

$$\begin{aligned}&\lim_{h\downarrow 0}\frac{\Pr(X^{(h)}=y\mid X^{(0)}=x)}{\Pr(X^{(h)}\neq x\mid X^{(0)}=x)}\\ =&\lim_{h\downarrow 0}\frac{\Pr(X^{(h)}=y\mid X^{(0)}=x)}{h}\Big/\frac{\Pr(X^{(h)}\neq x\mid X^{(0)}=x)}{h}\\ =&\lim_{h\downarrow 0}\frac{q_{x,y}h+o(h)}{h}\Big/\frac{q_x h+o(h)}{h}.\end{aligned}$$

We now find for the jump probability from state x to state y with $x\neq y$ that

$$\Pr(X^{(J_1)}=y\mid X^{(0)}=x)=\frac{q_{x,y}}{q_x}. \tag{3.19}$$

Finally, we will consider the joint probability distribution of jump times and jump probabilities

$$\Pr(X^{(J_1)}=y\wedge J_1>t\mid X^{(0)}=x). \tag{3.20}$$

We derive a "forward" equation for this probability, For $x,y\in S$, $t\in\mathbb{R}_{\geq 0}$, and $h>0$ we have

$$\begin{aligned}&\Pr(X^{(J_1)}=y\wedge J_1>t+h\mid X^{(0)}=x)\\ &=\sum_{z\in S}\Pr(X^{(J_1)}=y\wedge J_1>t+h\wedge X^{(h)}=z\mid X^{(0)}=x)\\ &=\Pr(X^{(J_1)}=y\wedge J_1>t+h\wedge X^{(h)}=x\mid X^{(0)}=x)\\ &=\Pr(X^{(J_1)}=y\wedge J_1>t+h\mid X^{(h)}=x\wedge X^{(0)}=x)\Pr(X^{(h)}=x\mid X^{(0)}=x).\end{aligned}$$

Above we used the fact that if $X^{(h)}$ is unequal to $X^{(0)}$, then the first jump-time must be smaller or equal to h. Applying the Markov property, homogeneity, and (3.8) we find

$$\begin{aligned}&\Pr(X^{(J_1)}=y\wedge J_1>t+h\mid X^{(0)}=x)\\ &\quad=\Pr(X^{(J_1)}=y\wedge J_1>t\mid X^{(0)}=x)(1-q_x h+o(h)).\end{aligned}$$

For the derivative of (3.20) we now find

$$\begin{aligned}&\frac{d}{dt}\Pr(X^{(J_1)}=y\wedge J_1>t\mid X^{(0)}=x)=\\ &\qquad-q_x\Pr(X^{(J_1)}=y\wedge J_1>t\mid X^{(0)}=x).\end{aligned} \tag{3.21}$$

Now, since x is stable we have that J_1 is always greater than zero. We then find, for the case $t=0$, that

$$\Pr(X^{(J_1)}=y\wedge J_1>0\mid X^{(0)}=x)=\Pr(X^{(J_1)}=y\mid X^{(0)}=x)=\frac{q_{x,y}}{q_x}.$$

This leads to the following solution to (3.21)

$$\Pr(X^{(J_1)}=y\wedge J_1>t\mid X^{(0)}=x)=\frac{q_{x,y}}{q_x}e^{-q_x t} \tag{3.22}$$

or

$$\Pr(X^{(J_1)} = y \wedge J_1 \leq t \mid X^{(0)} = x) = \frac{q_{x,y}}{q_x} - \frac{q_{x,y}}{q_x} e^{-q_x t}. \tag{3.23}$$

One importance consequence of (3.23) is that jump times and jump probabilities are independent since we have

$$\begin{aligned}\Pr(X^{(J_1)} &= y \wedge J_1 \leq t \mid X^{(0)} = x) \\ &= \Pr(X^{(J_1)} = y \mid X^{(0)} = x) \Pr(J_1 \leq t \mid X^{(0)} = x).\end{aligned}$$

As a final note on jump times and jump probabilities, we have that the probability to jump twice or more in a time-interval $t \in \mathbb{R}_{\geq 0}$ is $o(t)$.

3.1.4 Finite-jump probabilities

We have seen that the infinitesimal generator matrix Q of a stable Markov chain X determines the time at which X jumps and the next state to which X jumps. We will use this information to attempt to construct a transition function P from an infinitesimal generator matrix Q. Our construction follows the construction by Anderson [1]. The reason we repeat this construction in detail is that we will need to use similar constructions for our compositional models in Chapter 6.

Before we consider the transition function P, which describes how to move from one state to another with any number of jumps, we examine the probability to go from one state to another in a limited amount of jumps.

Definition 14 (n-jump probabilities)**.** *For $n \in \mathbb{N}$, let $P^{(n)}_{x,y}(t)$ be the probability that the Markov chain X reaches state y from state x at time t in at most n jumps,*

$$P^{(n)}_{x,y}(t) = \Pr(X^{(t)} = y \wedge J_{n+1} > t \mid X^{(0)} = x). \tag{3.24}$$

For $n = 0$ we have

$$P^{(0)}_{x,y}(t) = \Pr(X^{(t)} = y \wedge J_1 > t \mid X^{(0)} = x).$$

For $x \neq y$ we have that it is impossible to reach y from x in at most zero jumps. For $x = y$ we simply find the residence distribution (3.17), i.e., the probability to stay in x for at least t time-units.

$$P^{(0)}_{x,y}(t) = \begin{cases} 0 & \text{, if } x \neq y \\ e^{-q_x t} & \text{, if } x = y. \end{cases} \tag{3.25}$$

We now consider the case that $n > 0$. We study the derivative of $P^{(n)}_{x,y}$ to derive a "forward" equation. For any $h > 0$ we have

$$\begin{aligned}P^{(n)}_{x,y}(t+h) &= \Pr(X^{(t+h)} = y \wedge J_{n+1} > t+h \mid X^{(0)} = x) \\ &= \sum_{z \in S} \Pr(X^{(t+h)} = y \wedge J_{n+1} > t+h \wedge X^{(t)} = z \mid X^{(0)} = x) \\ &= \sum_{z \neq y} \Pr(X^{(t+h)} = y \wedge J_{n+1} > t+h \wedge X^{(t)} = z \mid X^{(0)} = x) \\ &\quad + \Pr(X^{(t+h)} = y \wedge J_{n+1} > t+h \wedge X^{(t)} = y \mid X^{(0)} = x).\end{aligned}$$

We now consider the number of jumps that may occur between t and $t+h$. The probability that two jumps or more occur within h time-units is $o(h)$. If at most one jump occurs between t and $t+h$ then, since $z \neq y$, the event

$$\{X^{(t+h)} = y \wedge J_{n+1} > t+h \wedge X^{(t)} = z\}$$

is equivalent to

$$\{X^{(t+h)} = y \wedge J_n > t \wedge X^{(t)} = z\}.$$

Similarly we have that

$$\{X^{(t+h)} = y \wedge J_{n+1} > t+h \wedge X^{(t)} = y\}$$

is equivalent to

$$\{X^{(t+h)} = y \wedge J_{n+1} > t \wedge X^{(t)} = y\}$$

when at most one jump occurs between t and $t+h$. Applying the above to $P^{(n)}_{x,y}(t+h)$ we find

$$\begin{aligned}
&\sum_{z\neq y} \Pr(X^{(t+h)} = y \wedge J_n > t \wedge X^{(t)} = z \mid X^{(0)} = x) \\
&\qquad + \Pr(X^{(t+h)} = y \wedge J_{n+1} > t \wedge X^{(t)} = y \mid X^{(0)} = x) + o(h) \\
&= \sum_{z\neq y} \Pr(X^{(t+h)} = y \mid J_n > t \wedge X^{(t)} = z \wedge X^{(0)} = x) \\
&\qquad\qquad \cdot \Pr(J_n > t \wedge X^{(t)} = z \mid X^{(0)} = x) \\
&\qquad + \Pr(X^{(t+h)} = y \mid J_{n+1} > t \wedge X^{(t)} = y \wedge X^{(0)} = x) \\
&\qquad\qquad \cdot \Pr(J_{n+1} > t \wedge X^{(t)} = y \mid X^{(0)} = x) + o(h).
\end{aligned}$$

The statement $J_{n+1} > t$ can be completely expressed in terms of the values of X between 0 and t. This means we can apply the Markov property above to find

$$\begin{aligned}
&\sum_{z\neq y} \Pr(X^{(t+h)} = y \mid X^{(t)} = z) P^{(n-1)}_{x,z}(t) \\
&\qquad + \Pr(X^{(t+h)} = y \mid X^{(t)} = y) P^{(n)}_{x,y}(t) + o(h)
\end{aligned}$$

We then apply the homogeneity of X and (3.8) to find

$$\begin{aligned}
&\sum_{z\neq y} (q_{z,y}h + o(h)) P^{(n-1)}_{x,z}(t) \\
&\qquad - (q_y h + o(h)) P^{(n)}_{x,y}(t) + P^{(n)}_{x,y}(t) + o(h).
\end{aligned}$$

We can apply the above result to the derivative of $P^{(n)}_{x,y}$ to find

$$\begin{aligned}
\frac{d}{dt} P^{(n)}_{x,y}(t) &= \lim_{h\downarrow 0} \frac{P^{(n)}_{x,y}(t+h) - P^{(n)}_{x,y}(t)}{h} \\
&= \lim_{h\downarrow 0} \frac{\sum_{z\neq y}(q_{z,y}h + o(h)) P^{(n-1)}_{x,z}(t) - (q_y h + o(h)) P^{(n)}_{x,y}(t) + o(h)}{h},
\end{aligned}$$

which leads to

$$\frac{d}{dt}P_{x,y}^{(n)}(t) = \sum_{z\neq y} q_{z,y}P_{x,z}^{(n-1)}(t) - q_y P_{x,y}^{(n)}(t). \tag{3.26}$$

The equation (3.26) is a first-order linear differential equation. It is important to note that, given the functions $P_{x,z}^{(n-1)}$ the equation (3.26) has only one unknown, namely $P_{x,y}^{(n)}$ in contrast to, for instance, (3.14) which may contain infinitely (and countably) many differential equations with as many variables. Using calculus we can solve (3.26) to find the unique solution

$$P_{x,y}^{(n)}(t) = e^{-q_y t}\left(P_{x,y}^{(n)}(0) + \int_0^t \sum_{z\neq x} P_{x,z}^{(n-1)}(s)q_{z,y}e^{-q_y s}ds\right).$$

For $t = 0$ we find

$$P_{x,y}^{(n)}(0) = \begin{cases} 0 & \text{, if } x \neq y \\ 1 & \text{, if } x = y. \end{cases} \tag{3.27}$$

Using (3.27) we then have

$$P_{x,y}^{(n)}(t) = \begin{cases} e^{-q_y t} + \int_0^t \sum_{z\neq y} P_{x,z}^{(n-1)}(s)q_{z,y}e^{-q_y(t-s)}ds & \text{, if } x = y \\ \int_0^t \sum_{z\neq y} P_{x,z}^{(n-1)}(s)q_{z,y}e^{-q_y(t-s)}ds & \text{, if } x \neq y. \end{cases} \tag{3.28}$$

Equation (3.28) can be rewritten as follows, where $\gamma_{x,y}$ equals one if $x = y$ and zero otherwise,

$$P_{x,y}^{(n)}(t) = \underbrace{\gamma_{x,y}e^{-q_y t}}_{\substack{x \text{ to } y \\ \text{in zero jumps}}} + \int_0^t \sum_{z\neq y} \underbrace{P_{x,z}^{(n-1)}(s)}_{\substack{x \text{ to } z \text{ within} \\ n-1 \text{ jumps}}} \underbrace{q_{z,y}}_{\substack{z \text{ to } y \text{ between} \\ s \text{ and } s+ds}} \underbrace{e^{-q_y(t-s)}}_{\substack{\text{stay in } y \\ \text{from } s+ds \text{ to } t}} ds$$

Note that $q_{z,y}$ is the derivative of (3.22) at time zero.

In a similar way we can also derive a "backward" version of (3.28) [1]

$$P_{x,y}^{(n)}(t) = \begin{cases} e^{-q_y t} + \int_0^t e^{-q_x(s)} \sum_{z\neq y} q_{x,z}P_{z,y}^{(n-1)}(t-s)ds & \text{, if } x = y \\ \int_0^t e^{-q_x(s)} \sum_{z\neq y} q_{x,z}P_{z,y}^{(n-1)}(t-s)ds & \text{, if } x \neq y. \end{cases} \tag{3.29}$$

This equation can be rewritten as follows,

$$P_{x,y}^{(n)}(t) = \underbrace{\gamma_{x,y}e^{-q_y t}}_{\substack{x \text{ to } y \\ \text{in zero jumps}}} + \int_0^t \underbrace{q_x e^{-q_x(s)}}_{\substack{\text{first jump} \\ \text{at } s}} \sum_{z\neq y} \underbrace{\frac{q_{x,z}}{q_x}}_{\substack{\text{jump} \\ \text{to } z}} \underbrace{P_{z,y}^{(n-1)}(s)}_{\substack{z \text{ to } y \text{ within} \\ n-1 \text{ jumps}}} ds$$

For a proof that (3.28) describes the same function as (3.29) we refer to Anderson [1].

We now define the *finite-jump* transition function f as the limit of $P^{(n)}$ when n goes to infinity.

Definition 15 (Finite-jump transition function)**.** *For states $x, y \in S$, we define the* finite-jump transition function $f_{x,y} : \mathbb{R}_{\geq 0} \to [0,1]$ *as the limit of the n-jump transition function* $P^{(n)}_{x,y}$

$$f_{x,y}(t) \equiv \lim_{n\to\infty} P^{(n)}_{x,y}(t).$$

The probability $f_{x,y}(t)$ is the probability that X reaches state y from x in a finite number of jumps.

Similarly, let J_∞ be the n-th jump-time where n goes to infinity,

Definition 16. *The* time of (first) explosion J_∞ *is the random variable*

$$J_\infty = \lim_{n\to\infty} J_n.$$

The random variable J_∞ is also called the *explosion time*. Note that J_∞ may take the value ∞.

We then have that

$$f_{x,y}(t) = \Pr(X^{(t)} = y \wedge J_\infty > t \mid X^{(0)} = x).$$

It is crucial to understand the relationship between the function $f_{x,y}$, which is defined uniquely by the infinitesimal generator matrix Q, and the transition function $P_{x,y}$ which (assuming it is stable) uniquely defines the behaviour of the Markov chain X. This relationship depends on the value of J_∞. If J_∞ can be finite, then X may perform infinitely many jumps in a finite amount of time. We say that X *explodes*. If J_∞ is infinite with probability one, then X cannot perform infinitely many jumps in a finite amount of time.

Proposition 1. *If the stable Markov chain X performs infinitely many jumps in a finite amount of time with probability zero, then*

$$P_{x,y}(t) = f_{x,y}(t),$$

for all $x, y \in S$ and $t \in \mathbb{R}_{\geq 0}$.

Proof. By the law of total probability we have

$$\begin{aligned} P_{x,y}(t) &= \Pr(X^{(t)} = y \mid X^{(0)} = x) \\ &= \Pr(X^{(t)} = y \wedge J_\infty > t \mid X^{(0)} = x) + \\ &\quad \Pr(X^{(t)} = y \wedge J_\infty \leq t \mid X^{(0)} = x) \end{aligned}$$

If we assume that X performs infinitely many jumps in a finite amount of time with probability zero then the probability that J_∞ is less than or equal to t is zero for all $t \in \mathbb{R}_{\geq 0}$. □

The function f also has the following important properties.

Proposition 2. *The finite-jump transition function f of a stable Markov chain X*

1. *is a standard transition function,*

2. *has derivative $f'(0) = Q$ at time zero,*

3. *is the minimal solution to the backward and forward equations of Q (for any other solution $\bar{f}(t)$ of the backward or forward equations we have $f_{x,y}(t) \leq \bar{f}_{x,y}(t)$ for all $x, y \in S$ and $t \in \mathbb{R}_{\geq 0}$), and*

4. *is the unique solution to the backward and forward equations of Q if $\sum_{y \in S} f_{x,y}(t) = 1$ for all $x \in S$ and $t \in \mathbb{R}_{\geq 0}$.*

For a proof of Proposition 2 we refer to Anderson [1]. The third item can be understood as follows. When a Markov chain reaches the time of first explosion, its behaviour is not specified by the infinitesimal generator matrix. We can then find infinitely many solutions to the backward and forward Kolmogorov equations by arbitrarily choosing the next state of the Markov chain when it explodes. The finite-jump transition function f represent the solution to the backward and forward equations where the Markov chain "stops" upon exploding. I.e., the "exploding" probability mass is not assigned to any state. All other solutions, must then be state-wise greater or equal to f, which means it is the minimal solution to the forward and backward equations.

The fourth item follows directly from Proposition 1. Let X be a Markov chain with infinitesimal generator matrix Q and let $f(t)$ be the transition function derived from Q. Then if the function $f(t)$ sums up to one, the probability that X explodes must be zero. By Proposition 1, it follows that the transition function $P(t)$ of any such Markov chain X must be equal to $f(t)$, but there can be only one transition function $P(t)$.

3.1.5 Regularity

If a Markov chain X performs infinitely many jumps in a finite amount of time with probability zero (i.e., it does not explode), then we say that X is *regular*. By item 4 of Proposition 2 we have seen that the regularity of X is determined by its infinitesimal generator matrix Q. The infinitesimal generator matrix Q of a regular Markov chain is also called regular. We now look at sufficient (and in some cases necessary) conditions on a matrix Q to be regular.

Before we consider Markov chains in general, we first have a look at *birth processes*. Given a sequence of birth rates $\{\lambda_i \in \mathbb{R}_{\geq 0} \mid i \in \mathbb{N}\}$ a birth process is a Markov chain with state space $\mathbb{N}$, where for all $i \in \mathbb{N}$ we have,

$$q_{i,j} = \begin{cases} \lambda_i, & \text{if } j = i + 1, \\ -\lambda_i, & \text{if } j = i, \\ 0, & \text{otherwise.} \end{cases}$$

In other words a birth process models a population where at each point in time an individual may be "born", increasing the population by one. The probability that a new individual is born depends on the current population. A nice property of birth processes is that their jump chain (the sequence of states the process jumps to) is completely fixed

by $X^{(J_i)} = i$ for each $i \in \mathbb{N}$. The expected n-th jump time is then simply the expected sum of n independent exponentially-distributed random variables,

$$E(J_n) = \sum_{i=0}^{n-1} 1/\lambda_i$$

and the expected time of explosion is

$$E(J_\infty) = \sum_{i=0}^{\infty} 1/\lambda_i. \tag{3.30}$$

This expected time of explosion is important because it is closely related to the probability of explosion. For any birth process we have that [1]

$$\Pr(J_\infty = \infty) = 1 \Leftrightarrow E(J_\infty) = \infty.$$

The regularity of a birth process then depends on whether the infinite sum on the right-hand side of (3.30) converges (in this case the birth process explodes with probability one) or diverges to infinity (in the case the birth process explodes with probability zero). For general Markov chains the situation is more difficult since the jump-chain is not completely fixed and the infinite sum in (3.30) may converge for certain paths and diverge for others. Still, we can already prove that Markov chains that are *uniformly bounded*, i.e., Markov chains whose exit-rates have a finite supremum, are regular.

Lemma 1 ([1])**.** *Let X be a stable Markov chain with state space S and infinitesimal generator Q. If Q is uniformly bounded, i.e.,*

$$\sup_{x \in S} q_x < \infty,$$

then X is regular.

Proof. Let $q_{\max}$ be the supremum exit-rate,

$$q_{\max} = \sup_{x \in S} q_x.$$

For any infinite sequence $\{x_i \in S \mid i \in \mathbb{N}\}$ of states we have

$$\sum_{i=0}^{\infty} 1/q_{x_i} \geq \sum_{i=0}^{\infty} 1/q_{\max} = \infty.$$

This means that regardless of the sequence of states X visits, the expected time of explosion is always infinite. □

An immediate consequence of Lemma 1 is that stable Markov chains with a finite state space are also regular.

Corollary 1 ([1])**.** *Let X be a stable Markov chain with state space S and infinitesimal generator Q. If S is finite then X is regular.*

Proof. Stability means that all exit-rates of states of X are finite and then the finiteness of S gives us that the supremum exit-rate must also be finite. Applying Lemma 1 then gives us that X is regular. □

Let's return to the study of regularity for birth processes and let's further assume that our birth process is not uniformly bounded. It turns out that a sufficient condition for the regularity of the birth process is the *linearity* of its birth-rates with respect to the population number. For any real-valued constant $c > 0$ we have

$$\sum_{i=1}^{\infty} \frac{1}{ci} = \infty,$$

but if the birth-rates grow faster than linearly we find

$$\sum_{i=1}^{\infty} \frac{1}{ci^{1+\epsilon}} < \infty,$$

for any real-valued constant $\epsilon > 0$. So, if there exists some constant $c > 0$ such that $q_i \leq ci$ for all $i \in \mathbb{N}$ then the birth process is regular.

We will now try to find regularity conditions for general Markov chains. Consider a stable Markov chain X with an arbitrary, countably infinite state space S and infinitesimal generator Q that is not uniformly bounded. Since the jump times of X depend on the states visited we cannot directly compute the expected time of explosion as we did for birth processes. Still, we can try to see if the exit-rate of X changes linearly by looking at the *expected change of exit-rate*. The exit-rate of X at time t is $q_{X^{(t)}}$. Let's consider the expected change of the exit-rate $q_{X^{(t)}}$ given that X occupies a particular state $x \in S$

$$\begin{aligned} E(\frac{d}{dt} q_{X^{(t)}} \mid X^{(t)} = x) &= \sum_{y \in S} (q_y - q_x) \lim_{h \to 0} \frac{\Pr(X^{(t+h)} = y \mid X^{(t)} = x)}{h} \\ &= \sum_{y \neq x} (q_y - q_x) q_{x,y}. \end{aligned}$$

It turns out that, to guarantee regularity, it is enough to show that this expected change of exit-rate is linear with respect to the current exit-rate. That is, if there exists some constant $c > 0$ such that for all states $x \in S$ we have

$$\sum_{y \neq x} (q_y - q_x) q_{x,y} \leq c q_x, \tag{3.31}$$

then X is regular.

In fact, the above result can be improved by considering the expected change of any function g over the states that goes to infinity as the exit-rates go to infinity.

Lemma 2 ([1]). *Let X be a stable Markov chain with state space S and infinitesimal generator Q, such that the exit-rates of X are not uniformly bounded. The Markov chain X is regular if there exists a series of subsets of S, $\{S_i \subset S \mid i \in \mathbb{N}\}$ and a function $g : S \to \mathbb{R}_{\geq 0}$, such that*

1. $\lim_{i\to\infty} S_i = S$,
2. $\sup_{x\in S_i} q_x < \infty$ *for all* $i \in \mathbb{N}$,
3. $\lim_{i\to\infty} \inf_{x\notin S_i} g(x) = \infty$, *and*
4. *there exists a constant* $c \in \mathbb{R}_{\geq 0}$ *such that*

$$\sum_{y\in S, y\neq x} (g(y) - g(x))q_{xy} \leq cg(x), \tag{3.32}$$

for all $x \in S$.

For the proof of Lemma 2 we refer to Anderson [1]. Intuitively, the first three conditions of Lemma 2 ensure that the function g goes to infinity as the exit-rates of X go to infinity. The first two conditions tell us that the exit-rates go to infinity as we traverse the state space along the series of subsets S_i (or rather their inverses $S \setminus S_i$). The third condition then gives us that the function g also goes to infinity as we move along these subsets. Finally, the fourth condition gives us that the expected change of g is linear with respect to the value of g for all states. Note, that we find (3.31) by choosing $g(x) = q_x$. We can then easily find appropriate subsets S_i, for instance, by choosing $S_i = \{x \in S \mid q_x \leq i\}$.

Example 7. *Consider a birth process X with birth-rates $\lambda_i = i$ for all $i \in \mathbb{N}$. We now find*

$$\sum_{y\neq x} (q_y - q_x)q_{x,y} = (i + 1 - i)i = i.$$

It follows that this birth process is regular.

Example 8. *To show that we sometimes need to choose the function g to be something other than the exit-rate, consider a Markov chain X with state space $\mathbb{N} \times \{0, 1\}$ and infinitesimal generator Q, where*

$$q_{x,y} = \begin{cases} i/2, & \text{if } x = (i, 0), y = (i + 1, 0) \text{ or } y = (i, 1), \\ i^2, & \text{if } i > 1, x = (i, 1), y = (i - 1, 1) \text{ or } x = (1, 1), y = (1, 0), \\ -i, & \text{if } x = y = (i, 0), \\ -i^2, & \text{if } x = y = (i, 1), \\ 0, & \text{otherwise.} \end{cases}$$

The states $(i, 0)$ from a linear birth process where with probability $1/2$ the Markov chain moves from state $(i, 0)$ to $(i, 1)$. The states $(i, 1)$ on the other hand form a quadratic death process. For a state $x = (i, 0)$ we find $q_x = i$ and

$$\sum_{y\neq x} (q_y - q_x)q_{x,y} = (i^2 - i)i/2 + (i + 1 - i)i/2 = 1/2(i^3 - i^2 + i).$$

Obviously there is no constant $c > 0$ such that $1/2(i^3 - i^2 + i) \leq ci$ for all $i \in \mathbb{N}$. However if we select subsets $S_i = \{(j,k) \mid j \leq i, k \in \{0,1\}\}$ and define the function $g : S \to \mathbb{R}_{\geq 0}$ such that $g(i,0) = g(i,1) = i$ then we find that the first three conditions of Lemma 2 hold and for the fourth condition we find for a state $x = (i,0)$ that

$$\sum_{y \neq x} (g(y) - g(x)) q_{x,y} = i/2$$

and for a state $x = (i,1)$, $i > 1$ we have

$$\sum_{y \neq x} (g(y) - g(x)) q_{x,y} = -i^2/2.$$

Finally we have for the state $x = (1,1)$ that the expected change of g is zero. We then have

$$\sum_{y \neq x} (g(y) - g(x)) q_{x,y} \leq 1/2 g(x)$$

for all states $x \in S$ and then X is regular according to Lemma 2.

We now summarise the above results in the following theorem which also gives several necessary and sufficient conditions for regularity.

Theorem 1 ([1])**.** *Given a stable infinitesimal generator matrix Q on a countable state space S,*

1. *Q is regular if,*
   (a) *S is finite,*
   (b) *Q is* uniformly *bounded, i.e.,*

   $$\sup_{x \in S} q_x < +\infty,$$

   or,
   (c) *there exist a series of subsets of S, $(S_i, i \in \mathbb{N}, i \geq 1)$ and a function $g : S \to \mathbb{R}_{\geq 0}$, such that*
   - $\lim_{i \to \infty} S_i = S$,
   - $\sup_{x \in S_i} q_x < +\infty$ *for all* $i \geq 1$,
   - $\lim_{i \to \infty} \inf_{x \notin S_i} g(x) = +\infty$, *and*
   - *there exists a constant $c \in \mathbb{R}_{\geq 0}$ such that*

     $$\sum_{y \in S} q_{xy} g(y) \leq c g(x),$$

     for all $x \in S$.
2. *Q is regular if and only if,*

(a) *the finite-jump transition function f is the unique solution of the backward Kolmogorov equation,*

(b) *the equation $Qv = \lambda v, 0 \leq v \leq 1$, i.e.,*

$$\sum_{y \in S} q_{x,y} v_y = \lambda v_x, \qquad 0 \leq v_x \leq 1, x \in S, \tag{3.33}$$

has no non-trivial solution for some $\lambda > 0$,

(c) *the inequality $Qv \geq \lambda v, 0 \leq v \leq 1$, i.e.,*

$$\sum_{y \in S} q_{x,y} v_y \geq \lambda v_x, \qquad 0 \leq v_x \leq 1, x \in S, \tag{3.34}$$

has no non-trivial solution for some $\lambda > 0$,

(d) *the equation $Qv = \lambda v, -1 \leq v \leq 1$, i.e.,*

$$\sum_{y \in S} q_{x,y} v_y = \lambda v_x, \qquad -1 \leq v_x \leq 1, x \in S, \tag{3.35}$$

has no non-trivial solution for some $\lambda > 0$,

(e) *the finite-jump transition function f is the unique solution of the forward Kolmogorov equation, or*

(f) *the equation $vQ = \lambda v, v \geq 0, \sum_{y \in S} v_y < \infty$, i.e.,*

$$\sum_{x \in S} v_x q_{x,y} = \lambda v_y, \qquad v_x \geq 0, y \in S, \tag{3.36}$$

has no non-trivial solution for some $\lambda > 0$ such that $\sum_{y \in S} v_y < \infty$.

The proof of Theorem 1 can be found in Anderson [1]. For the necessary and sufficient conditions 2a and 2e we have that the finite-jump transition function f is the *minimal* solution to the backward respectively forward Kolmogorov equations. For the Kolmogorov equations there is at least one solution g such that $\sum_{x \in S} g_x = 1$ [1]. Then, if f is the only solution to either of the Kolmogorov equations we have $\sum_{x \in S} f(x) = 1$, which means the probability of explosion must be zero. It follows that the corresponding infinitesimal generator must be regular. For condition 2f it is useful to interpret the vector v as a transient distribution of some Markov chain X with infinitesimal generator Q. The left-hand side of (3.36) is then just (3.13) and we can rewrite it to

$$\frac{d}{dt}\pi_y = \lambda \pi_y,$$

where $\pi_y = \Pr(X^{(t)} = y)$ for some arbitrary time-point t. Now, since $\lambda > 0$ this would mean that the derivative of the probability distribution is positive for all states and strictly positive for at least one (since we exclude the trivial solution $v = 0$). It turns out such a mysterious probability distribution only exists when the Markov chain X is not regular.

3.1.6 Sufficient conditions for the Markov property

So far we have studied the properties of Markov chains. We have seen that, if a Markov chain is stable and regular, its probabilistic behaviour is completely determined by its infinitesimal generator matrix. However, these results only hold when we know the stochastic process in question is in fact a Markov chain. In this subsection we consider what conditions a stochastic process must fulfil such that it is a Markov chain.

In order to be a Markov chain, a stable jump process must of course have a, possibly time-dependent, infinitesimal generator function $Q(t)$. However, this is not sufficient to guarantee that it satisfies the Markov property. A sufficient condition for the Markov property is that the stable jump process has the "Markov property up to $o(h)$" and is non-explosive[2].

We will consider a stable jump-process $\{X^{(t)}, t \in \mathbb{R}_{\geq 0}\}$, which takes values on a state space S and which has the follow "Markov property up to $o(h)$. For any distinct pair of states $y, x \in S$ and time-point $t \in \mathbb{R}_{\geq 0}$ we find a constant $q_{x,y}^{(t)}$ such that for any states $x_1, \ldots, x_n \in S$, and time-points $t + h > t > t_1 > \ldots > t_n \in \mathbb{R}_{\geq 0}$ we have

$$\begin{aligned} &\Pr(X^{(t+h)} = y \mid X^{(t)} = x, X^{(t_1)} = x_1, \ldots, X^{(t_n)} = x_n) \\ &= q_{x,y}^{(t)} h + o(h). \end{aligned} \tag{3.37}$$

As usual we denote the jump times of X by $J_0, \ldots$. Additionally we denote the i-th jump-time after time t as

$$J_i^{(t)} = \begin{cases} t, & \text{if } i = 0, \\ \inf\{s > J_{i-1}^{(t)} \mid X^{(s)} \neq X^{(J_{i-1}^{(t)})}\}. \end{cases}$$

First of all, we can immediately derive from (3.37) that

$$\begin{aligned} &\Pr(X^{(t+h)} = x \mid X^{(t)} = x, X^{(t_1)} = x_1, \ldots, X^{(t_n)} = x_n) \\ &= 1 - q_x^{(t)} h + o(h), \end{aligned} \tag{3.38}$$

where

$$q_x^{(t)} = \sum_{y \neq x} q_{x,y}^{(t)}.$$

Furthermore, we find that the probability to make two jumps in a time-interval $[t, t+h]$ is small, i.e.,

$$\Pr(J_2^{(t)} < t + h \mid X^{(t)} = x) = o(h).$$

We will now derive a forward equation in the style of (3.28) using (3.37). As a first step, we compute the residence time of a state x given a particular history. For states $x, x_1, \ldots, x_n \in S$, time-points $s, t > t_1 > \ldots > t_n \in \mathbb{R}_{\geq 0}$ we write

$$E_{x,H}^{(s)}(t) \equiv \Pr(J_1^{(s)} > s + t \mid X^{(s)} = x, H),$$

[2]The following Theorem appears to be used implicitly in many papers. However, we could not find it in the literature.

where the *history* H denotes the event

$$\{X^{(t_1)} = x_1, \ldots, X^{(t_n)} = x_n\}.$$

Lemma 3. *For any state x, any history H, and any time-points t, s we have*

$$E^{(s)}_{x,H}(t) = e^{-\int_s^{s+t} q_x^{(t')} dt'}. \tag{3.39}$$

Proof. In a similar way as we derived (3.16) for Markov chains we can now derive the following differential equation for $E^{(s)}_{x,H}(t)$, namely

$$\frac{d}{dt} E^{(s)}_{x,H}(t) = -q_x^{(s+t)} E^{(s)}_{x,H}(t). \tag{3.40}$$

We also have

$$E^{(s)}_{x,H}(0) = 1.$$

Then (3.39) is the unique solution to (3.40). □

We proceed by determining the finite jump transition probabilities of our jump process recursively, as we have done for Markov chains. For states $y, x, x_1, \ldots, x_n \in S$, time-points $s, t > t_1 > \ldots > t_n \in \mathbb{R}_{\geq 0}$, and index $n \in \mathbb{N}$ we write

$$P^{(n)}_{x,y,H}(s,t) \equiv \Pr(X^{(s+t)} = y, J^{(s)}_n > s + t \mid X^{(s)} = x, H),$$

for the probability to occupy state y at time $t + s$ starting in state x at time s, given history

$$H = \{X^{(t_1)} = x_1, \ldots, X^{(t_n)} = x_n\}$$

with at most n jumps in the time-interval $[s, s + t]$.

Lemma 4. *For any states x, y, any history H, any time-points t, s, and any jump-index $n \in \mathbb{N}$ we have*

$$\begin{aligned} &P^{(n)}_{x,y,H}(s,t) \\ &\quad = \psi_{x,y} e^{-\int_0^t q_y^{(s+t')} dt'} + \int_0^t \sum_{z \neq y} q_{z,y}^{(s+t')} P^{(n-1)}_{x,z,H}(s,t') e^{-\int_{t'}^t q_y^{(s+t'')} dt''} dt', \end{aligned} \tag{3.41}$$

where

$$\psi_{x,y} = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{if } x \neq y. \end{cases}$$

Proof. In a similar way as we derived (3.26) for Markov chains we can now derive the following differential equation for $P^{(n)}_{x,y,H}(s,t)$, namely

$$\frac{d}{dt} P^{(n)}_{x,y,H}(s,t) = \sum_{z \neq y} q_{z,y}^{(s+t)} P^{(n-1)}_{x,z,H}(s,t) - q_y^{(s+t)} P^{(n)}_{x,y,H}(s,t). \tag{3.42}$$

We also have

$$P^{(n)}_{x,y,H}(s,0) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{if } x \neq y. \end{cases}$$

Then (3.41) is the unique solution to (3.42). □

Theorem 2. *If the stable jump process X satisfies (3.37) and it is non-explosive, i.e.,*

$$\Pr(J_\infty = \infty) = 1, \tag{3.43}$$

then X is a Markov chain with generator function $Q(t)$ as in (3.37) and (3.38).

Proof. Since X is non-explosive we have for any pair of states $x, y \in S$, any time-points $s, t \in \mathbb{R}_{\geq 0}$, and any two histories H, H', that

$$\begin{aligned} &\Pr(X^{(s+t)} = y \mid X^{(s)} = x, H) \\ &= \Pr(X^{(s+t)} = y, J^{(s)}_\infty > s + t \mid X^{(s)} = x, H) \\ &= \lim_{n\to\infty} P^{(n)}_{x,y,H}(s,t) \\ &= \lim_{n\to\infty} P^{(n)}_{x,y,H'}(s,t) \\ &= \Pr(X^{(s+t)} = y \mid X^{(s)} = x, H'). \end{aligned}$$

The Markov property follows by choosing H' to be the empty history. The fact that X has infinitesimal generator function $Q(t)$ follows directly from (3.37) and (3.38). □

3.2 Bisimulation

In this section we will discuss when stable Markov chains can be considered equivalent. We will consider two Markov chains X and $\mathbf{X}$ equivalent, when their transient state probabilities are equal. That is, X and $\mathbf{X}$ are equivalent, if and only if for all states $s \in S_{\mathsf{all}}$ and time-points $t \in \mathbb{R}_{\geq 0}$

$$\Pr(X^{(t)} =_s x) = \Pr(\mathbf{X}^{(t)} =_s x).$$

However, we relax this condition to finite-jump probabilities, i.e.,

$$\Pr(X^{(t)} =_s x, J_\infty > t) = \Pr(\mathbf{X}^{(t)} =_s x, J_\infty > t).$$

This restriction allows us to concentrate on infinitesimal generator matrices, since the finite-jump transition probabilities are completely defined by the initial distribution and infinitesimal generator matrix of a stable Markov chain. Before finding an equivalence relation for infinitesimal generator matrices we will first define an equivalence relation on the states of a single Markov chain. Our goal is to find an equivalence relation that satisfies, for any pair of equivalent states x, y and any other state $z \in S_{\mathsf{all}}$ that

$$\Pr(X^{(t)} =_s z, J_\infty > t \mid X^{(t)} = x) = \Pr(X^{(t)} =_s z, J_\infty > t \mid X^{(t)} = y)$$

for any time-point $t \in \mathbb{R}_{\geq 0}$. It turns out we can accomplish this by defining a *bisimulation relation* on the states of a Markov chain. Two states are then equivalent if they can simulate each others infinitesimal jump probabilities. The bisimulation relation we discuss was first introduced as *lumpability* or *ordinary lumpability* [8] for finite-state discrete-time Markov chains and has later been extended to continuous-time Markov chains. We use the name bisimulation instead of lumpability to reflect the similarity to bisimulation relations as used in automata theory. Our contribution is the study of bisimulation relation for infinite-state, possibly irregular Markov chains. We will show that, even for irregular Markov chains, the finite-jump transient distributions are preserved by bisimulation up to $=_s$.

3.2.1 Basic definition

In the following we consider a stable Markov chain X with state space S and infinitesimal generator matrix Q.

Definition 17 ([8])**.** *An equivalence relation $\mathcal{E}$ on S is a* bisimulation with respect to *Q if, for each pair of states $x\mathcal{E}y$ in S and each equivalence class D in $S/\mathcal{E}$, we have that*

$$x \notin D \implies \sum_{z \in D} q_{x,z} = \sum_{z \in D} q_{y,z}. \tag{3.44}$$

When clear from context which infinitesimal generator is meant, we will call $\mathcal{E}$ a bisimulation*.*

We will be especially interested in bisimulations that refine the state-equivalence $=_s$, but we will develop the theory for arbitrary bisimulations. It is worth mentioning that the universal equivalence relation on S (that equates all states) is trivially a bisimulation for any infinitesimal generator matrix.

From the definition of bisimulation we find that, for any two states x and y in an equivalence class D of $\mathcal{E}$, their cumulative infinitesimal transition probabilities to states outside D is the same

$$\sum_{z \notin D} q_{x,z} = \sum_{D' \neq D} \sum_{z \in D'} q_{x,z} = \sum_{D' \neq D} \sum_{z \in D'} q_{y,z} = \sum_{z \notin D} q_{y,z}.$$

We write $\bar{q}_D$ for this cumulative infinitesimal transition probability,

$$\bar{q}_D = \sum_{z \notin D} q_{x,z},$$

for an arbitrary state $x \in D$. Similarly we write $\bar{q}_{D,D'}$ for the cumulative infinitesimal transition probability to reach an equivalence class D' from D,

$$\bar{q}_{D,D'} = \sum_{z \in D'} \bar{q}_{x,z}, \tag{3.45}$$

for an arbitrary state $x \in D$. We then define the $|S/\mathcal{E}| \times |S/\mathcal{E}|$ matrix $\bar{Q}$ with entries $\{\bar{q}_{D,D'} \mid D, D' \in S/\mathcal{E}\}$, where

$$\bar{q}_{D,D} = -\bar{q}_D. \tag{3.46}$$

Note that all row-sums of $\bar{Q}$ are zero, all diagonal entries are negative and finite, and all off-diagonal entries are positive and finite.

We will now lift the concept of jump times and jump probabilities to the equivalence classes of a bisimulation $\mathcal{E}$. Our goal is to show that they can be derived from $\bar{Q}$, in the same way as the normal jump times and probabilities of X are derived from Q. We will see that this is only the case under certain conditions.

3.2.2 Jump times and jump probabilities

In the following we consider an equivalence relation $\mathcal{E}$ on S which is a bisimulation with respect to Q. We are interested in showing that this equivalence relation somehow preserves the jump times jump probabilities of the Markov chain. However, we will see that this is only the case "up to" $\mathcal{E}$. That is, the probabilities to jump from one *equivalence* class of $\mathcal{E}$ to another are preserved by $\mathcal{E}$.

First, we consider the jump times between equivalence classes. We define jump times for equivalence classes recursively as we did for the state-based jump times

$$\bar{J}_n = \begin{cases} 0 & \text{, if } n = 0 \\ \inf\{t \mid t > \bar{J}_{n-1}, X^{(t)} \notin D, X^{(\bar{J}_{n-1})} \in D\} & \text{, if } n > 0. \end{cases}$$

We now consider the first jump-time, or residence time, $\bar{J}_1$ of an equivalence class D. From the above definition we have

$$\bar{J}_1 = \inf\{t \mid X^{(t)} \notin D, \text{ where } D \text{ is such that } X^{(0)} \in D\}.$$

We consider the distribution of $\bar{J}_1$ conditioned on the starting state of X and define the cumulative distribution function $E_x : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for $x \in D$ as the probability that $\bar{J}_1$ is smaller or equal to a time-point t under the condition that X starts in x.

$$E_x(t) = P(\bar{J}_1 \leq t \mid X^{(0)} = x).$$

Lemma 5. *Given a bisimulation $\mathcal{E}$ on S with respect to Q, let D be an equivalence class of $\mathcal{E}$. Then*

1. *the cumulative distributions functions $E_x(t)$, $x \in D$ satisfy, for all $t \in \mathbb{R}_{\geq 0}$,*

$$\frac{d}{dt}E_x(t) = \bar{q}_D + \sum_{y \in D} q_{x,y} E_y(t), \quad 0 \leq E_x(t) \leq 1, x \in D \tag{3.47}$$

and

2. *the Laplace transforms $r_x(s)$ of $E_x(t)$, $x \in D$ satisfy, for all $s > 0$,*

$$s r_x(s) = \frac{\bar{q}_D}{s} + \sum_{y \in D} q_{x,y} r_y(s), \quad 0 \leq r_x(s) \leq 1, x \in D. \tag{3.48}$$

Proof. Since $E_x(t)$ is a distribution function it follows that $0 \leq E_x(t) \leq 1$ for all $x \in D$ and $t \in \mathbb{R}_{\geq 0}$. It then also follows that $0 \leq r_x(s) \leq 1$ for all $x \in D$ and $s > 0$.

1. We consider first the probability $E_x(t+h)$ for $t \in \mathbb{R}_{\geq 0}$ and $h > 0$,

$$\begin{aligned} E_x(t+h) &= \Pr(\bar{J}_1 \leq t+h \mid X^{(0)} = x) \\ &= \sum_{y \in S} \Pr(\bar{J}_1 \leq t+h \wedge X^{(h)} = y \mid X^{(0)} = x) \\ &= \sum_{y \in D} \Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x) \cdot \\ &\qquad \Pr(X^{(h)} = y \mid X^{(0)} = x) + \\ &\qquad \sum_{y \notin D} \Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x) \cdot \\ &\qquad \Pr(X^{(h)} = y \mid X^{(0)} = x). \end{aligned}$$

We would like to apply the Markov property to the probability $\Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x)$ for $x, y \in D$, but we cannot do so directly as the event $\{\bar{J}_1 \leq t+h\}$ also describes the Markov chain X before time-point h. However, we find

$$\begin{aligned} &\Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x) \\ &\quad = 1 - \Pr(\bar{J}_1 > t+h \mid X^{(h)} = y \wedge X^{(0)} = x) \\ &\quad = 1 - \frac{\Pr(\bar{J}_1 > t+h \wedge X^{(h)} = y \wedge X^{(0)} = x)}{\Pr(X^{(h)} = y \wedge X^{(0)} = x)} \\ &\quad = 1 - \frac{\Pr(\{X^{(t')} \in D \mid 0 \leq t' \leq t+h\} \wedge X^{(h)} = y \wedge X^{(0)} = x)}{\Pr(X^{(h)} = y \wedge X^{(0)} = x)}. \end{aligned} \tag{3.49}$$

Given the fact that the probability of more than two jumps (between states) occurring in the time-interval $[0, h]$ is $o(h)$ we have

$$\begin{aligned} &\Pr(\{X^{(t')} \in D \mid 0 \leq t' \leq t+h\} \wedge X^{(h)} = y \wedge X^{(0)} = x) \\ &= \Pr(\{X^{(t')} \in D \mid 0 \leq t' \leq t+h\} \wedge J_2 > h \wedge X^{(h)} = y \wedge X^{(0)} = x) + o(h) \end{aligned}$$

Now, the fact that X is in D at time-points 0 and h, and at most one jump (between states) occurred in this time period implies that X was in D for the whole interval $[0, h]$. We then find

$$\begin{aligned} &\Pr(\{X^{(t')} \in D \mid 0 \leq t' \leq t+h\} \wedge J_2 > h \wedge X^{(h)} = y \wedge X^{(0)} = x) \\ &\quad = \Pr(\{X^{(t')} \in D \mid h \leq t' \leq t+h\} \wedge J_2 > h \wedge X^{(h)} = y \wedge X^{(0)} = x) \\ &\quad = \Pr(\{X^{(t')} \in D \mid h \leq t' \leq t+h\} \wedge X^{(h)} = y \wedge X^{(0)} = x) - o(h). \end{aligned}$$

Note that for two $o(h)$ functions $f(h)$ and $g(h)$ we have that the function $(f(h) - g(h))/\Pr(X^{(h)} = y \wedge X^{(0)} = x)$ is also $o(h)$. Applying the above to (3.49) we find

that we can indeed apply the Markov property and use the homogeneity of X to find,

$$\begin{aligned}
&\Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x) \\
&\quad = 1 - \Pr(\{X^{(t')} \in D \mid h \leq t' \leq t+h\} \mid X^{(h)} = y \wedge X^{(0)} = x) + o(h) \\
&\quad = 1 - \Pr(\{X^{(t')} \in D \mid h \leq t' \leq t+h\} \mid X^{(h)} = y) + o(h) \\
&\quad = 1 - \Pr(\{X^{(t')} \in D \mid 0 \leq t' \leq t\} \mid X^{(0)} = y) + o(h) \\
&\quad = \Pr(\bar{J}_1 \leq t \mid X^{(0)} = y) + o(h)
\end{aligned}$$

For states $y \notin D$ we trivially have that $\Pr(\bar{J}_1 \leq t+h \mid X^{(h)} = y \wedge X^{(0)} = x)$ equals one. We then find

$$\begin{aligned}
E_x(t+h) &= \sum_{y \in D} (\Pr(\bar{J}_1 \leq t \mid X^{(0)} = y) + o(h)) \Pr(X^{(h)} = y \mid X^{(0)} = x) \\
&\qquad + \sum_{y \notin D} \Pr(X^{(h)} = y \mid X^{(0)} = x) \\
&= \sum_{y \in D, y \neq x} (E_y(t) + o(h))(q_{x,y}h + o(h)) \\
&\qquad + (E_x(t) + o(h))(1 - q_x h + o(h)) + \sum_{y \notin D} (q_{x,y}h + o(h)) \\
&= \sum_{y \in D} q_{x,y} h E_y(t) + \sum_{y \notin D} q_{x,y} h + E_x(t) + o(h). \qquad \boxed{3.50}
\end{aligned}$$

We now consider the derivative of $E_x(t)$ and derive a "backward" equation:

$$\frac{d}{dt} E_x(t) = \lim_{h \downarrow 0} \frac{E_x(t+h) - E_x(t)}{h}.$$

We apply $\boxed{3.50}$ and find that $\frac{d}{dt} E_x(t)$ equals

$$\lim_{h \downarrow 0} \frac{\sum_{y \in D} q_{x,y} h E_y(t) + \sum_{y \notin D} q_{x,y} h + o(h)}{h}.$$

Since $\sum_{y \notin D} q_{x,y} = \bar{q}_D$, we have

$$\frac{d}{dt} E_x(t) = \bar{q}_D + \sum_{y \in D} q_{x,y} E_y(t).$$

2. We apply the Laplace transform to $\boxed{3.47}$ to find

$$s r_x(s) - E_x(0) = \frac{\bar{q}_D}{s} + \sum_{y \in D} q_{x,y} r_y(s).$$

We trivially have that $E_x(0) = 0$ for all $x \in D$ which means that the equations $\boxed{3.48}$ must hold for the Laplace transforms of $E_x(t)$, for each $x \in D$.

□

Lemma 6. *The negative exponential distributions*

$$E_x(t) = 1 - e^{-\bar{q}_D t}, \text{ for } x \in D \tag{3.51}$$

form a solution to (3.47).

Proof. For the Laplace transform of (3.51) we find

$$r_x(s) = \frac{\bar{q}_D}{s(s+\bar{q}_D)}, \text{ for } x \in D. \tag{3.52}$$

We now prove Lemma 6 by showing that, for all $s > 0$, (3.52) is a solution to (3.48). Substituting the former into the latter gives us

$$\begin{aligned}\frac{\bar{q}_D s}{s(s+\bar{q}_D)} &= \frac{\bar{q}_D}{s} + \sum_{y \in D} q_{x,y} \cdot \frac{\bar{q}_D}{s(s+\bar{q}_D)} \\ &= \frac{\bar{q}_D}{s} + \frac{\bar{q}_D}{s(s+\bar{q}_D)} \sum_{y \in D} q_{x,y}.\end{aligned}$$

From the fact that the rows of Q add up to zero we then find $\sum_{y \in D} q_{x,y} = -\sum_{z \notin D} q_{x,z} = -\bar{q}_D$. We can then rewrite the above to

$$\frac{\bar{q}_D s}{s(s+\bar{q}_D)} = \frac{\bar{q}_D}{s} - \frac{\bar{q}_D^2}{s(s+\bar{q}_D)}.$$

Since the above holds for all $s > 0$ we have that (3.52) is indeed a solution to (3.48) and then $E_x(t) = 1 - e^{-\bar{q}_D t}$, for $x \in D$ is a solution to (3.47). □

Lemma 6 seems to give us the expected result, that the residence time of an equivalence class of a bisimulation relation is exponentially distributed with parameter $\bar{q}_D$. However, the exponential distribution (3.51) may not be the *only* solution to (3.47). In fact, this is only the case for "regular" equivalence classes. We call an equivalence class D regular if the infinitesimal generator matrix Q projected onto D is regular. Of course, this projection is in general not an infinitesimal generator matrix itself as its rows do not sum up to zero. We circumvent this problem by adjoining an absorbing state to D[3].

Let Q' be the infinitesimal generator matrix obtained by adding an absorbing state $\perp$ to the equivalence class D, i.e.,

$$Q' = \begin{pmatrix} Q[D] & \bar{q}_D \mathbf{1} \\ \mathbf{0} & 0 \end{pmatrix}. \tag{3.53}$$

[3]This construction is necessary as we restrict our attention to so-called *conservative* infinitesimal generator matrices. Markov chains with non-conservative infinitesimal generator matrices (i.e., with non-zero row-sums) have also been studied in the literature [1].

Here $Q[D]$ is the infinitesimal generator matrix Q restricted to the equivalence class D, $\mathbf{1}$ is a column-vector of size $|D|$ containing only ones, and $\mathbf{0}$ is a row-vector of size $|D|$ containing only zeroes. We have

$$q'_{x,y} = \begin{cases} q_{x,y} & \text{, if } x, y \in D \\ \bar{q}_D & \text{, if } x \in D, y = \perp \\ 0 & \text{, otherwise.} \end{cases}$$

for all $x, y \in D \cup \{\perp\}$.

Lemma 7. *If Q' is regular, then (3.52) is the unique non-trivial solution to (3.47).*

Proof. We prove Lemma 7 by contradiction. Assume that there are two distinct non-trivial solutions $E_x(t)$, $x \in D$ and $\bar{E}_x(t)$, $x \in D$ to (3.47). These solutions then have distinct Laplace transforms $r_x(s)$, $x \in D$ and $\bar{r}_x(s)$, $x \in D$ that satisfy (3.48), i.e.,

$$sr_x(s) = \frac{\bar{q}_D}{s} + \sum_{y \in D} q_{x,y} \cdot r_y(s), \text{ and}$$
$$s\bar{r}_x(s) = \frac{\bar{q}_D}{s} + \sum_{y \in D} q_{x,y} \cdot \bar{r}_y(s).$$

Subtracting the above equations we have

$$s(r_x(s) - \bar{r}_x(s)) = \sum_{y \in D} q_{x,y} \cdot (r_y(s) - \bar{r}_y(s)). \tag{3.54}$$

Define the series of vectors $\{\mathbf{v}(s) \mid s > 0\}$ on $D \cup \{\perp\}$ as follows:

$$v_x(s) = \begin{cases} (r_x(s) - \bar{r}_x(s)) & \text{, if } x \in D, \\ 0 & \text{, if } x = \perp. \end{cases}$$

We then find from (3.54) and the definition of Q' that

$$sv_x(s) = \sum_{y \in D} q'_{x,y} \cdot v_y(s). \tag{3.55}$$

for $x \in D \cup \{\perp\}$.

Since $0 \leq r_x(s), \bar{r}_x(s) \leq 1$, for all $x \in D$, we have $-1 \leq v_x(s) \leq 1$, for all $x \in D \cup \{\perp\}$ and $s > 0$. Furthermore the distinctness of $r(s)$, $x \in D$ and $\bar{r}(s)$, $x \in D$ means that, $\mathbf{v}(s)$ is not the zero-vector. It follows that (3.55) is a counter-example to condition 2d of Theorem 1 and then Q' is not regular. This is a contradiction with our assumption that Q' is regular. □

We say that an equivalence class D is regular, if the infinitesimal generator matrix Q' defined in (3.53) is regular. Let $E_D(t)$ be the residence time distribution of D,

$$E_D(t) = \Pr(\bar{J}_1 \leq t \mid X^{(0)} \in D).$$

We now have that, if the equivalence class D is regular, then the residence time of D is exponentially distributed with rate $\bar{q}_D$.

Theorem 3. *If the equivalence class D is regular then the residence time of D is exponentially distributed with rate $\bar{q}_D$,*

$$E_D(t) = 1 - e^{-\bar{q}_D t}. \tag{3.56}$$

Proof. We find

$$\begin{aligned} E_D(t) &= P(\bar{J}_1 \leq t \mid X^{(0)} \in D) \\ &= \sum_{x \in D} P(\bar{J}_1 \leq t \wedge X^{(0)} = x \mid X^{(0)} \in D) \\ &= \sum_{x \in D} P(\bar{J}_1 \leq t \mid X^{(0)} = x) \cdot P(X^{(0)} = x \mid X^{(0)} \in D) \end{aligned}$$

We now apply Lemma 7 to find

$$\begin{aligned} E_D(t) &= \sum_{x \in D} (1 - e^{-\bar{q}_D t}) \cdot P(X^{(0)} = x \mid X^{(0)} \in D) \\ &= (1 - e^{-\bar{q}_D t}) \cdot \sum_{x \in D} P(X^{(0)} = x \mid X^{(0)} \in D) \end{aligned}$$

and then (3.56) follows. □

For the jump probabilities between equivalence classes we find a similar result as for the jump probabilities between states.

Theorem 4. *Given distinct equivalence classes D_1 and D_2, such that D_1 is regular, we have*

$$\Pr(X^{(\bar{J}_1)} \in D_2 \mid X^{(0)} \in D_1) = \frac{\bar{q}_{D_1,D_2}}{\bar{q}_{D_1}}.$$

The proof of Theorem 4 follows the derivation of (3.19).

Because X is time-homogeneous, we find that the above results also hold for later jump times. We now turn our attention to the finite-jump transition probabilities for the equivalence classes of $\mathcal{E}$.

3.2.3 Finite jump transition probabilities

We now show that two states x and y that are related by the bisimulation $\mathcal{E}$ have the same finite-jump transition probabilities, under the condition that the equivalence classes of the bisimulation are all regular. Let $\bar{J}_\infty$ denote the time of first explosion for the derived Markov chain Y,

$$\bar{J}_\infty = \lim_{n \to \infty} \bar{J}_n.$$

Theorem 5. *Given a bisimulation relation $\mathcal{E}$ on S, such that all equivalence classes of $\mathcal{E}$ are regular, we find that for two states $x\mathcal{E}y$, an equivalence class D and a time-point $t \in \mathbb{R}_{\geq 0}$, we have*

$$\Pr(X^{(t)} \in D \wedge \bar{J}_\infty > t \mid X^{(0)} = x) = \Pr(X^{(t)} \in D \wedge \bar{J}_\infty > t \mid X^{(0)} = y). \tag{3.57}$$

Proof. Given regular equivalence classes $D, \bar{D}$, a state $x \in \bar{D}$, and a natural number n, let $\bar{P}^{(n)}_{x,D}(t)$ be the probability that the Markov chain X reaches equivalence class D from state x in at most n jumps between equivalence classes,

$$\bar{P}^{(n)}_{x,D}(t) \equiv \Pr(X^{(t)} \in D \wedge \bar{J}_{n+1} > t \mid X^{(0)} = x)$$

In the following we will refer to jumps between equivalence classes simply as jumps. Whenever we discuss jumps between states of X, we will state so explicitly. We can now rewrite (3.57) in terms of $\bar{P}$,

$$\lim_{n\to\infty} \bar{P}^{(n)}_{x,D}(t) = \lim_{n\to\infty} \bar{P}^{(n)}_{y,D}(t). \tag{3.58}$$

We now show by recursion on n that for all $n \in \mathbb{N}$ we have

$$\bar{P}^{(n)}_{x,D}(t) = \bar{P}^{(n)}_{y,D}(t). \tag{3.59}$$

For the base case $n = 0$ we have

$$\bar{P}^{(0)}_{x,D}(t) = \Pr(X^{(t)} \in D \wedge \bar{J}_1 > t \mid X^{(0)} = x).$$

For $x \notin D$ we have that it is impossible to reach D from x in at most zero jumps. For $x \in D$ we simply find the residence distribution, i.e., the probability to stay in D for t time-units. Since D is regular we can apply Lemma 7 to find

$$\bar{P}^{(0)}_{x,D}(t) = \begin{cases} 0 & \text{, if } x \notin D \\ e^{-\bar{q}_D t} & \text{, if } x \in D. \end{cases}$$

The same holds for state y as it occupies the same equivalence class as x, and then

$$\bar{P}^{(0)}_{x,D}(t) = \bar{P}^{(0)}_{y,D}(t).$$

We now consider the case that $n > 0$. As our induction assumption we assume that (3.59) holds for $n-1$, i.e.,

$$\bar{P}^{(n-1)}_{x',D'}(t') = \bar{P}^{(n-1)}_{y',D'}(t'), \tag{3.60}$$

for all states $x'\mathcal{E}y'$, any equivalence class D' of $\mathcal{E}$, and any time-point $t' \in \mathbb{R}_{\geq 0}$.

We can derive a forward equation for the derivative of $\bar{P}^{(n)}$ in the same way as we did for $P^{(n)}$ (see the derivation of (3.26)) to find

$$\frac{d}{dt}\bar{P}^{(n)}_{x,D}(t) = \sum_{D' \neq D} q_{D',D} \bar{P}^{(n-1)}_{x,D'}(t) - \bar{q}_D \bar{P}^{(n)}_{x,D}(t). \tag{3.61}$$

The equation (3.61) is a first-order linear differential equation. For $t = 0$ we find

$$\bar{P}^{(n)}_{x,D}(0) = \begin{cases} 0 & \text{, if } x \notin D \\ 1 & \text{, if } x \in D. \end{cases} \tag{3.62}$$

We can then solve (3.61) to find

$$\bar{P}^{(n)}_{x,D}(t) = \begin{cases} e^{-\bar{q}_D t} + \int_0^t \sum_{D' \neq D} \bar{P}^{(n-1)}_{x,D'}(s) q_{D',D} e^{-\bar{q}_D(t-s)} ds & \text{, if } x \in D \\ \int_0^t \sum_{D' \neq D} \bar{P}^{(n-1)}_{x,D'}(s) q_{D',D} e^{-\bar{q}_D(t-s)} ds & \text{, if } x \notin D. \end{cases} \tag{3.63}$$

Similarly we find for state y that

$$\bar{P}^{(n)}_{y,D}(t) = \begin{cases} e^{-\bar{q}_D t} + \int_0^t \sum_{D' \neq D} \bar{P}^{(n-1)}_{y,D'}(s) q_{D',D} e^{-\bar{q}_D(t-s)} ds & \text{, if } y \in D \\ \int_0^t \sum_{D' \neq D} \bar{P}^{(n-1)}_{y,D'}(s) q_{D',D} e^{-\bar{q}_D(t-s)} ds & \text{, if } y \notin D. \end{cases} \tag{3.64}$$

Since x and y are equivalent according to $\mathcal{E}$, they must occupy the same equivalence class. So if x is in D then y is in D and vice versa. Furthermore, the induction assumption gives us that $\bar{P}^{(n-1)}_{x,D'}(s)$ equals $\bar{P}^{(n-1)}_{y,D'}(s)$ for all $s \in \mathbb{R}_{\geq 0}$. It follows that (3.63) is equal to (3.64) and then (3.57) holds. □

As we saw in Subsection 3.1.4 we can rewrite (3.63) as follows, where $\gamma_{x,D}$ equals one if $x \in D$ and zero otherwise,

$$\bar{P}^{(n)}_{x,D}(t) = \underbrace{\gamma_{x,D} e^{-\bar{q}_D t}}_{\substack{x \text{ to } D \\ \text{in zero jumps}}} + \int_0^t \sum_{D' \neq D} \underbrace{\bar{P}^{(n-1)}_{x,D'}(s)}_{\substack{x \text{ to } D' \text{ within} \\ n-1 \text{ jumps}}} \underbrace{q_{D',D}}_{\substack{D' \text{ to } D \text{ between} \\ s \text{ and } s+ds}} \underbrace{e^{-\bar{q}_D(t-s)}}_{\substack{\text{stay in } D \\ \text{from } s+ds \text{ to } t}} ds.$$

Corollary 2. *Given equivalence classes D, D' and time-point $t \in \mathbb{R}_{\geq 0}$ we have*

$$\Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} \in D) = \Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} = x) \tag{3.65}$$

for an arbitrary state $x \in D$.

Proof. We apply Theorem 5 to find

$$\begin{aligned} &\Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} = D) \\ &\quad = \sum_{y \in D} \Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \wedge X^{(0)} = y \mid X^{(0)} = D) \\ &\quad = \sum_{y \in D} \Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} = y) \Pr(X^{(0)} = y \mid X^{(0)} = D) \\ &\quad = \Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} = x) \sum_{y \in D} \Pr(X^{(0)} = y \mid X^{(0)} = D) \\ &\quad = \Pr(X^{(t)} \in D' \wedge \bar{J}_\infty > t \mid X^{(0)} = x). \end{aligned}$$

□

3.2.4 The quotient process

Given a bisimulation $\mathcal{E}$ on S and a state $x \in S$, let $[x]_{\mathcal{E}}$ denote the equivalence class which contains x, i.e.,

$$[x]_{\mathcal{E}} = \{y \mid x\mathcal{E}y\}.$$

We now define the *quotient process* as the process which moves from equivalence class to equivalence class as the Markov chain X moves from state to state.

Definition 18. *Given a bisimulation $\mathcal{E}$ on S, the* quotient process *of X with respect to $\mathcal{E}$ is a stochastic process $\{Y^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$, with state space $S/\mathcal{E}$ such that:*

$$Y^{(t)} = \left[X^{(t)}\right]_{\mathcal{E}}.$$

In the previous subsections we have already shown several properties of Y. Critically, we can now show that Y has the Markov property.

Theorem 6. *When X is regular, then for equivalence classes $D_0, \ldots, D_n$, and D and increasingly large time-points $t_0 < \ldots < t_n < t$ we have*

$$\Pr(Y^{(t)} = D \mid Y^{(t_n)} = D_n \wedge \ldots \wedge Y^{(t_0)} = D_0) = \Pr(Y^{(t)} = D \mid Y^{(t_n)} = D_n). \quad (3.66)$$

Proof. Since X is regular, we have that X jumps infinitely often in finite time with probability zero. As a jump of process Y can only occur when X jumps we have that Y is also "regular" in this sense. It is then enough to show that

$$\begin{aligned}\Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid Y^{(t_n)} = D_n \wedge \ldots \wedge Y^{(t_0)} = D_0)\\ = \Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid Y^{(t_n)} = D_n).\end{aligned}$$

The left-hand side of this equation is equal to

$$\begin{aligned}\sum_{y \in D_n} \Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid X^{(t_n)} = y \wedge Y^{(t_{n-1})} = D_{n-1} \wedge \ldots \wedge Y^{(t_0)} = D_0) \cdot\\ \Pr(X^{(t_n)} = y \mid Y^{(t_n)} = D_n \wedge \ldots \wedge Y^{(t_0)} = D_0).\end{aligned}$$

We can now apply the Markov property of X to find

$$\begin{aligned}\sum_{y \in D_n} \Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid X^{(t_n)} = y) \cdot\\ \Pr(X^{(t_n)} = y \mid Y^{(t_n)} = D_n \wedge \ldots \wedge Y^{(t_0)} = D_0).\end{aligned}$$

Let x be an arbitrary state in D_n we then apply Theorem 5 and the homogeneity of X to find the above equals

$$\begin{aligned}\Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid X^{(t_n)} = x) \cdot\\ \sum_{y \in D_n} \Pr(X^{(t_n)} = y \mid Y^{(t_n)} = D_n \wedge \ldots \wedge Y^{(t_0)} = D_0)\end{aligned}$$

which is trivially equal to

$$\Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid X^{(t_n)} = x).$$

Now we apply Corollary 2 and the homogeneity of X to find

$$\Pr(Y^{(t)} = D \wedge \bar{J}_\infty > T \mid Y^{(t_n)} = D_n).$$

□

We also have that Y is time-homogeneous.

Theorem 7. *When X is regular, then for equivalence classes D_1, D_2 and time-points t_1, t_2 we have*

$$\Pr(Y^{(t_1+t_2)} = D_2 \mid Y^{(t_1)} = D_1) = \Pr(Y^{(t_2)} = D_2 \mid Y^{(0)} = D_1). \tag{3.67}$$

Theorem 7 follows directly from the homogeneity of X. Now we are ready to show that Y is indeed a Markov chain. Moreover, it has infinitesimal generator matrix $\bar{Q}$ as defined in (3.45) and (3.46).

Theorem 8. *If X is regular, then Y is a regular, stable, time-homogeneous Markov chain with state space $S/\mathcal{E}$ and infinitesimal generator matrix $\bar{Q}$ which has entries $\{\bar{q}_{D_1,D_2} \mid D_1, D_2 \in S/\mathcal{E}\}$, where, for an arbitrary state $x \in D_1$ we have*

$$\bar{q}_{D_1,D_2} = \begin{cases} \sum_{y \in D_2} q_{x,y} & , \text{ if } D_1 \neq D_2, \\ -\sum_{y \notin D_1} q_{x,y} & , \text{ if } D_1 = D_2. \end{cases} \tag{3.68}$$

Proof. The regularity of Y follows directly from the regularity of X. We have shown that Y has the Markov property and is time-homogeneous. The fact that Y has infinitesimal generator matrix $\bar{Q}$ follows from a comparison of the unique (since Y is regular) finite-jump transition function of Y (see (3.63), this equation also holds when we replace x by D) and the finite-jump transition function derived from $\bar{Q}$ (see (3.28)). It is obvious that they are identical. Note that the transition function that has infinitesimal generator matrix $\bar{Q}$ must be unique, otherwise Y would not be regular. Finally, the stability of Y follows from the fact that, for an equivalence class D which contains state x, we have

$$\bar{q}_D = \sum_{y \notin D} q_{x,y}$$

and, since X is stable, $\sum_{y \neq x} q_{x,y}$ is finite for any state $x \in S$. □

3.2.5 Bisimulation for irregular Markov chains

Above, we have shown results for the case that X is regular. The case where X is not regular is more difficult to handle and we do not consider it in detail. However, we note that, if at least all equivalence classes of $\mathcal{E}$ are regular, then we can still construct a finite-jump transition function as in Subsection 3.2.3.

Example 9. *State space* $S = \mathbb{N} \times \{0, 1\}$. *We have*

$$q_{x,y} = \begin{cases} (x+1)^2, & \text{if } x = (i,j), y = (i+1,j) \\ 1, & \text{if } x = (i,0), y = (i,1) \\ 0, & \text{otherwise.} \end{cases}$$

Now the relation

$$\mathcal{E} = \{(x,y) \mid i, i' \in \mathbb{N}, j \in \{0,1\}, x = (i,j), y = (i',j)\}$$

is clearly a bisimulation. However, the two equivalence classes $D_0 = \{(i,0) \mid i \in \mathbb{N}\}$ *and* $D_1 = \{(i,1) \mid i \in \mathbb{N}\}$ *are* not *regular. Consider now a Markov chain* Y *with state space* $\{0,1\}$ *and infinitesimal generator matrix*

$$Q' = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}.$$

This is the Markov chain that we might expect to find as the quotient process of X *with respect to* $\mathcal{E}$. *However, we find*

$$\Pr(Y^{(t)} = D_1 \mid Y^{(0)} = D_0) = 1 - e^{-t},$$

but, since X *may explode within* t *time-units with probability greater than zero, it is possible that*

$$\Pr(X^{(t)} \in D_1 \wedge J_\infty > t \mid X^{(0)} \in D_0) < 1 - e^{-t}.$$

Of course, the actual transition probabilities of X *are not uniquely determined by its infinitesimal generator matrix. This shows that Theorem 8 does not hold for bisimulations with irregular equivalence classes.*

However, if the equivalence classes of the bisimulation are all regular, we find the following corollary to Theorem 5. Note that the proof of this theorem never uses the fact that the Markov chain in question is regular.

Corollary 3. *Given an irregular Markov chain* X *with infinitesimal generator matrix* Q *and a bisimulation* $\mathcal{E}$ *on the states of* X *such that the equivalence classes of* $\mathcal{E}$ *are regular, let* f *be the finite-jump transition function for* Q *and let* f' *be the finite-jump transition function for the infinitesimal generator function* Q' *on* $S/\mathcal{E}$ *with*

$$q'_{D,D'} = \sum_{y \in D'} q_{x,y}$$

for equivalence classes D, D' *and an arbitrary state* $x \in D$. *We then have*

$$f'_{D,D'}(t) = \sum_{y \in D'} f_{x,y}(t),$$

for any state $x \in D$.

In conclusion, a bisimulation with regular equivalence classes always "preserves" the finite-jump transition probability function of a Markov chain, even if it is irregular.

3.3 Discussion

In this chapter we have studied continuous-time Markov chains. We have focused, in the footsteps of Anderson [1], on the *finite-jump probabilities* of Markov chains, i.e., the probability of reaching a certain state with a certain finite number of jumps (recall Definition 14). The reason we have focused on the finite-jump probabilities is that they will play a central role when we study the jump processes that underlie I/O-IMCs in Chapter 6. We have also looked at the notion of *bisimulation* for Markov chains, giving a new proof for the soundness of bisimulation which is again based on the finite-jump probabilities of a Markov chain. This new proof is interesting for two reasons. First, it lays the groundwork for a similar proof for bisimulation on I/O-IMCs given in Chapter 7. Secondly, it shows, for the first time, under which conditions bisimulation can be soundly applied to infinite-state and even irregular Markov chains.

3.3.1 CTMCs as graph-based models

As discussed in this chapter, a CTMC is a jump-process that satisfies the Markov property. However, for regular CTMCs we find that they can be uniquely represented by their infinitesimal generator matrix. This matrix can of course in turn be represented as a graph with states as its vertices (if we restrict ourselves to countable state spaces) and edges labelled with the entries of the matrix. For a CTMC with state space S and generator matrix Q we then find the graph (S, E) with

$$E = \{(x, \lambda, y) \mid x, y \in S, x \neq y, q_{xy} = \lambda > 0\}$$

as its representation. In a sense, we can consider this graph to be the *syntax* of the CTMC, while the underlying jump process is the semantics of the graph. We will use this graph-based representation of a CTMC as one ingredient when constructing the graph-based representation of I/O-IMCs in Chapter 5. Similarly, we will use jump processes as inspiration for the semantics of I/O-IMCs in Chapter 6.

3.3.2 Composition of CTMCs

This thesis discusses a compositional Markov model, I/O-IMCs. It then makes sense to briefly consider what composition may mean in the context of CTMCs. It turns out there is one natural way to compose CTMCs (see for instance Hermanns and Zhang [26]). The idea is to compose two CTMCs by assuming they are completely independent. In terms of the graph representation of CTMCs this can be achieved by *interleaving* the two graphs. That is, we compose the graphs in a completely orthogonal way. Consider two graphs (S, E) and (S', E') which represent two CTMCs $\{X^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$ and $\{Y^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$. The composition of the two graphs is then the graph (S'', E'') with $S'' = S \times S'$ and

$$\begin{aligned} E'' = &\{((x, y), \lambda, (x', y)) \mid (x, \lambda, x') \in E, y \in S'\} \\ &\cup \{((x, y), \lambda, (x, y')) \mid (y, \lambda, y') \in E', x \in S\}. \end{aligned}$$

The semantics of this graph is exactly the independent combination of the two CTMCs: $\{Z^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$ where $Z^{(t)} = (X^{(t)}, Y^{(t)})$. We will leave it to the reader to prove that $Z^{(t)}$ is indeed the semantics of the graph (S'', E'') and that the semantics of our graph-based CTMC representations is thus modular.

As discussed in Chapter 1, this kind of composition is not very interesting, as the two CTMCs X and Y do not actually influence each other or interact in any way. Still, we will see that this composition is an important ingredient for the way we compose I/O-IMCs. In Chapter 5 we will see that the composition rules for the graph-representation of I/O-IMCs follows the composition of CTMCs we have just discussed (for the Markovian aspect of I/O-IMCs). Similarly, we will see in Chapter 6 that on the semantical level, the Markovian jumps induced by I/O-IMCs are composed by assuming they are completely independent.

4

Input/Output Automata

In this chapter we discuss a variant of *input/output automata* (IOA), a formalism introduced in 1989 by Lynch and Tuttle for the modelling and analysis of reactive systems [33]. IOA allow us to model the interactions between components, such as sub-algorithms of a distributed algorithm. Each IOA in a composition models a set of possible sequences of events called *traces*. Interaction is modelled through the fact that composed IOA must agree on the order in which these events occur. In other words, common events must be *synchronised*. One of the most important aspects of IOA is that their trace-based semantics is *sound* with respect to parallel composition. Any fair trace of a composite IOA can be projected onto its component IOA to find fair traces of these IOA.

This thesis uses IOA concepts to support the modelling of interactions between components in a compositional Markov modelling framework. Some variations to the original IOA theory are needed, because states – albeit viewed through state rewards or state labels – are decisive for Markov models. In contrast, the original IOA theory is mostly oblivious to the notion of a state, it instead develops a modular trace-based theory. Central to our approach is the fact that we are primarily interested in *state reachability* properties instead of only *trace observation* properties.

Contribution. In redeveloping the theory with adaptations as motivated above, we assure that the main properties of IOA, including the modularity of (fair) executions and traces, do extend smoothly to our setting. In particular, we establish that reach-trace equivalence (a variation on trace-equivalence) is the coarsest congruence (with respect to parallel composition and hiding) on IOA that preserves reachability properties. We also introduce weak bisimulation for IOA, which is strictly finer than reach-trace equivalence, as well as confluence to prepare for subsequent matters.

4.1 Basic Definition

This section gives the basic definition of IOA and briefly discusses its building blocks.

Definition 19 (Adapted from [33].)**.** *An input/output automaton (IOA) is a tuple* $P = \langle S, A, R^I, \hat{x} \rangle$*, where*

- *the* state space $S \subset S_{\mathsf{all}}$ *is a non-empty countable set,*
- *the* set of actions A *is a finite set partitioned into* input actions A^I*,* output actions A^O*, and internal or hidden actions* A^H*.*
- *the* interactive transition relation R^I *is a subset of* $S \times A \times S$*, and*
- *the initial state* $\hat{x}$ *is a member of* S*.*

We require that the IOA is input-enabled*; for any state* $x \in S$ *and any input action* $a \in A^I$ *there exists a state* $y \in S$ *such that there is a transition* $(x, a, y) \in R^I$*. We also require that the IOA is* finitely branching*, i.e., for all* $x \in S$ *we have* $|\{(x, a, y) \in R^I\}| < \infty$*.*

We will use the letters x, y, z as well as their indexed versions to range over states. The letters $a, b, c, \ldots$ will be used to range over actions. Whenever it is clear from context which IOA is meant, we will use the predicate $x \xrightarrow{a} y$ to denote the existence of a transition (x, a, y) in R^I.

Definition 19 differs from the original definition of IOA [33] in two ways. First, we fix a single starting state instead of allowing a set of starting states. Secondly, we do not partition the locally-controlled actions into *tasks*. The first change is made to simplify the discussion in this chapter, since it turns out we do not need to use multiple starting states. The second change is due to the fact that we will use a stronger notion of fairness.

Example 10. *As an example, Figure 4.1 shows an IOA model* P_{RC} *of a generic repairable component. It might be a processor in a computer system, a pump in a reactor cooling system, or a tire on a car. We only model the* failure behaviour *of the component. We model how the component may break down and how it may subsequently be repaired. We have* $P_{RC} = \langle S, A, R^I, \mathbf{failing} \rangle$ *with as its state space* $S = \{\mathbf{failing}, \mathbf{down}, \mathbf{recovering}, \mathbf{up}\}$*, output actions* $A^O = \{fail, recover\}$*, input actions* $A^I = \{repair\}$*, and internal actions* $A^H = \emptyset$*. The transitions in* R^I *are described by Figure 4.1. Ellipses denote the states of the component. Double arrows denote the transitions (throughout this thesis single arrows are used for "Markovian" transitions, e.g., transitions in a CTMC and double arrows for "interactive" transitions, e.g., transitions in an IOA). The actions of transitions are embellished with a question-mark when the action is an input action, an exclamation mark when it is an output action and a semicolon if the action is an internal action. The arrow with no source ending in state* **failing** *identifies it as the starting state. In the remainder we will assume that, unless explicitly noted, for any action of an IOA we find at least one transition labelled with this action. In this way, an IOA is completely defined by its graph representation.*

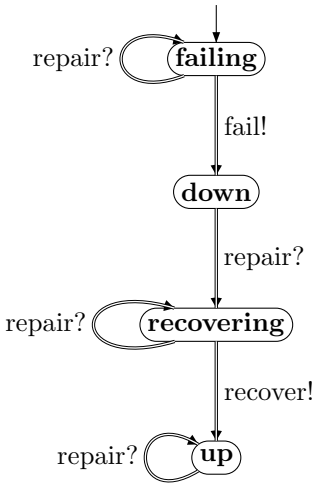


Figure 4.1: Example of an IOA.

In the remainder of this chapter we will consider an IOA $P = \langle S, A, R^I, x\rangle$ unless otherwise specified. To the IOA P we can associate a directed, edge-labelled, graph with vertices S, labels A, and edges R^I. When we talk about the paths of P, we mean the paths of the graph associated with P. We refer to the set $A^I \cup A^O$ as the *visible* actions of P. The actions in the set $A^O \cup A^H$ are called *locally controlled*. If the set of input actions A^I is empty, we call the IOA P *closed* and if the set of visible actions is empty the IOA is *complete*. Given a finite sequence of objects $\sigma = s_1, \ldots, s_n$ and an object s_{n+1} we will write $\sigma \circ s_{n+1}$ for the *concatenation* of σ with s_{n+1}, i.e., $\sigma \circ s_{n+1} = s_1, \ldots, s_n, s_{n+1}$. We also allow concatenation of two sequences in the obvious way. Given a set B we will write $\sigma{\downarrow}B$ for the *projection* of σ onto B, i.e.,

$$\sigma{\downarrow}B = \begin{cases} s_1 \circ s_2, \ldots, s_n{\downarrow}B, & \text{if } s_1 \in B, \\ s_2, \ldots, s_n{\downarrow}B, & \text{if } s_1 \notin B. \end{cases}$$

Throughout, we will use ϵ to denote an empty sequence.

Definition 20. *For a action a, we say a transition $(x, a, y) \in R^I$ is* enabled *in state x for IOA P. The locally-controlled action a is enabled in x for P if there exists a state y such that $(x, a, y) \in R^I$. We lift enabledness to sets by saying that a set of transitions $R \subset R^I$ is enabled in a state x if one of its constituent transitions is enabled in x. Similarly, a set of actions $B \subset A^O \cup A^H$ is enabled in a state x if one of the actions in B is enabled in x. We denote the set of all actions enabled in a state x for IOA P as $En_P(x)$ and leave out the subscript whenever the IOA is clear from context.*

4.2 Classification of states

This section reviews two ways of classifying the states of IOA based on their possible behaviour. These classifications will prove to be useful throughout the thesis when

discussing states of an IOA.

The first way of distinguishing states looks at the presence of enabled transitions in a state. We will see that this concept is closely related to the maximal progress assumption, which states that whenever *any* transition is enabled that cannot be delayed, *some* transition will occur instantaneously [39]. The name "maximal progress assumption" originates in formalisms that combine timed transitions with instantaneous transitions, where instantaneous transitions are assumed to take precedence over timed (or delayed transitions). However, a similar assumption appears in the context of IOA as a fairness assumption (see Section 4.4) [33]. For IOA, the transitions that cannot be delayed are the locally-controlled transitions. This means that, when an IOA occupies a state with outgoing output or internal transitions it may not tarry in this state and must leave it by any transition (which may also be an input transition). We will use the name "maximal progress assumption" also to refer to the fairness assumption for IOA. We call states with outgoing locally-controlled transition *unstable*. A state that is not unstable is called *stable*.

Definition 21. *A state $x \in S$ is* unstable *for P if any locally-controlled action is enabled in x. A state $x \in S$ is stable for P if it is not unstable. We use the predicate $st_P(x)$ to denote that x is stable for P and leave out the subscript whenever the IOA is clear from context.*

Recall, that we also used the notion of *stable states* in the context of CTMCs to describe states that have a finite exit-rate. To avoid confusion we will from now on use the term stability only in the context of IOA or I/O-IMCs. By construction, all CTMCs that we consider will contain only stable states.

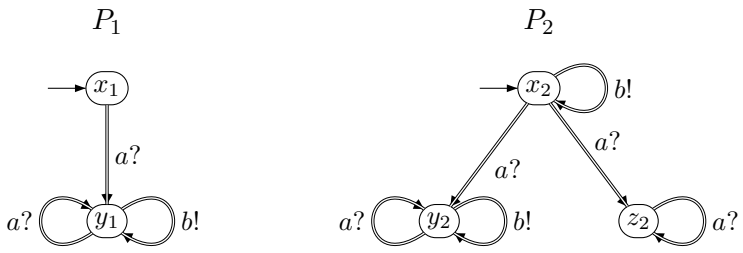
Example 11. *The IOA P_{RC} from Example 10 has stable states* **down** *and* **up**. *States* **failing** *and* **recovering** *are unstable.*

An interesting situation arises when an IOA occupies a state that can never reach a stable state. In this case the IOA is forced to move, without time passing, from state to state indefinitely. We call this situation, in which time is not allowed to progress, time divergence.

Definition 22. *A state $x \in S$ is* divergent *for P if there exists no path in P starting in x and ending in a stable state. We use the predicate $div_P(x)$ to denote that a state x is divergent and leave out the subscript whenever the IOA is clear from context.*

We now give an example of IOA with divergent states.

For IOA P_1 in Figure 4.2 we have that state x_1 is stable as its only outgoing transition is an input transition. However, state y_1 is both unstable and divergent as it has an outgoing output action and there is no path from y_1 to x_1. For IOA P_2 we have that state z_2 is stable, state y_2 is unstable and divergent, and state x_2 is neither stable nor divergent.



Example 12.

Figure 4.2: Two IOAs with stable and divergent states.

It should be noted that time divergence is not the same as explosiveness in CTMCs. Time divergence occurs when infinitely many interactive transitions occur *instantaneously* (i.e., without time passing). Explosion in a CTMC occurs when infinitely many transitions occur in a finite amount of time. Specifically, explosion occurs when transitions occur at ever decreasing intervals, in such a way that the series of transition times converges.

4.3 Executions, Traces, and Reachability

We can now define the key elements of the semantics of IOA. The linear-time semantics of an IOA is described in terms of *executions*, *traces*, and *reachable states*. Executions are paths in the IOA. The trace of an execution is the sequence of visible actions that appear along the execution. A finite execution also provides proof that its final state is *reachable*. We will treat time-divergence explicitly and consider two different kinds of executions (and traces), *non-divergent* (ND) executions and *explicitly divergent* (ED) executions. Non-divergent executions and traces are identical to executions and traces as defined by Lynch and Tuttle [33].

4.3.1 Executions

Definition 23 ([33])**.** *Given an IOA P with states S, a non-divergent* execution fragment *of P is a, possibly infinite, path σ in P. That is, σ is either a finite sequence $x_0, a_0, x_1, a_1, \ldots, x_{n-1}a_{n-1}x_n$ such that for all $0 \leq i < n$, $(x_i, a_i, x_{i+1}) \in R^I$ or an infinite sequence $x_0, a_0, x_1, a_1, \ldots.$ such that for all $i \in \mathbb{N}$, $(x_i, a_i, x_{i+1}) \in R^I$. For a state $x \in S$ we denote the set of all execution fragments of P starting in x (i.e., $x_0 = x$) as $NDEx_P(x)$. We leave out the subscript whenever the IOA is clear from context.*

An execution *is an execution fragment that starts in the initial state of P. We denote the set of all executions of P as $NDEx(P)$ and we have $NDEx(P) = NDEx_P(\hat{x})$.*

We will model time divergence explicitly using a distinguished state $\perp$ which is not in the state space of any IOA and which only appears in its executions and behaviours.

Whenever an execution ends in $\perp$, this represents the fact that time divergence has occurred. Time divergence may occur due to the IOA reaching a divergent state itself and then being forced to perform infinitely many transitions without time passing. This is called *local time divergence*. However, time divergence may also occur due to another IOA reaching a divergent state. This is called *external time divergence*. Since divergence may be external, any execution may end, at any time, in the explicit-divergence state $\perp$. The executions in which divergence is made explicit are called *explicit-divergence executions* or ED-executions. The associated traces are called ED-traces.

Definition 24. *An* explicit-divergence execution fragment *(ED-execution fragment) of IOA P is a finite execution fragment of P extended by the explicit-divergence state $\perp$. For a state $x \in S$ we denote the set of all ED-execution fragments of P starting in x as $EDEx_P(x)$. We leave out the subscript whenever the IOA is clear from context. An explicit-divergence execution of P is an ED-execution fragment that starts in the initial state of P. We denote the set of all ED-executions of P, $EDEx(P)$ and find*

$$EDEx(P) = \{\sigma \circ \langle \perp \rangle \mid \sigma \in NDEx(P), |\sigma| < \infty\}.$$

The shortest ED-execution of any IOA is $\langle \hat{x}, \perp \rangle$, where $\hat{x}$ is the initial state. This ED-execution has length zero as it contains zero transitions.

We say an ED-execution is *locally* divergent if the last state (before $\perp$) is divergent. Otherwise, we say that the ED-execution is *externally* divergent. Intuitively, if an IOA follows an externally divergent execution, it means some other IOA performs a locally divergent execution.

Definition 25. *The set of all execution fragments of P starting in a state x is denoted $Ex_P(x)$. We leave out the subscript whenever the IOA is clear from context. Similarly, the set of all executions of the IOA P is denoted $Ex(P)$. We have $Ex_P(x) = NDEx_P(x) \cup EDEx_P(x)$ and $Ex(P) = NDEx(P) \cup EDEx(P)$.*

4.3.2 Traces

It is important to have a notion of what part of an execution *influences* other IOA or *is influenced by* other IOA. The part of an execution that is used to communicate between IOA is called the *trace* and consists of the visible actions appearing along the execution.

Definition 26. *Given an execution fragment σ of P, the* trace *of σ, denoted $Tr(\sigma)$, is the sequence of visible actions $\sigma \downarrow (A^O \cup A^I)$ along σ. A trace is non-divergent respectively explicitly divergent if the associated execution fragment is non-divergent respectively explicitly divergent. We denote the set of non-divergent traces of a state x for P as $NDTr_P(x)$, the set of explicit-divergence traces of a state x is denoted $EDTr_P(x)$. The set of all traces of a state x is denoted $Tr_P(x) = NDTr_P(x) \cup EDTr_P(x)$. We have $Tr_P(x) = \{Tr(\sigma) \mid \sigma \in Ex_P(x)\}$. We leave out the subscripts whenever the IOA is clear from context. The set of non-divergent, explicit-divergence, respectively all traces of P are, denoted $NDTr(P)$, $EDTr(P)$, respectively $Tr(P)$ and are the non-divergent, explicit-divergent, respectively all traces of the initial state of P.*

4.3.3 Reachable states

In the original treatment of IOA, the most important aspect of an IOA is its set of traces. However, since we use IOA to describe the interactions that may occur in between Markovian transitions, we focus on *reachability* properties.

Definition 27. *Given two states $x, y \in S$, we say y is* reachable *from x in P if x has a finite execution σ for P such that $last(\sigma) = y$. The explicit divergence state $\perp$ is, by definition, reachable from any state. We denote the set of all states reachable from x for P as $Reach_P(x)$ and leave out the subscript when clear from context. A state is reachable in P if it is reachable from the initial state of P. We write $Reach(P)$ for the set of all reachable states in P.*

4.3.4 Reach-trace

From a finite execution we can derive both a *trace*, the visible actions along the execution and a *reachable state*, the final state of the execution. We will see that this combination of trace and final state of an execution is important to our treatment of IOA. In essence, the trace contains all the information needed to compose IOA, and the state reached by an execution is the information that is visible by an outside observer.

Definition 28. *Given a finite execution fragment σ, the* reach-trace *of σ is the pair $\langle w, y\rangle$ where w is the trace of σ and y is the final state (which may be $\perp$) of σ. We denote the set of all reach-traces of a state x for P as $RT_P(x)$ and leave out the subscript when clear from context. The set of all reach-traces of an IOA P, denoted $RT(P)$, is the set of all reach-traces of its initial state.*

Example 13. *The IOA P_{RC} from Example 10 has, among others, the following executions.*

$\langle$**failing**$\rangle$,
$\langle$**failing**, *fail*, **down**$\rangle$,
$\langle$**failing**, *fail*, **down**, $\perp\rangle$,
$\langle$**failing**, *fail*, **down**, *repair*, **recovering**, *repair*, **recovering**$\rangle$,
$\langle$**failing**, *fail*, **down**, *repair*, **recovering**, *recover*, **up**$\rangle$, *and*
$\langle$**failing**, *repair*, **failing**, *repair*, **failing**, *repair*, **failing**, $\ldots\rangle$.

The first five executions are finite; the associated reach-traces are respectively

$(\epsilon$, **failing**$)$,
$(\langle$*fail*$\rangle$, **down**$)$,
$(\langle$*fail*$\rangle, \perp)$
$(\langle$*fail*, *repair*, *repair*$\rangle$, **recovering**$)$, *and*
$(\langle$*fail*, *repair*, *recover*$\rangle$, **up**$)$.

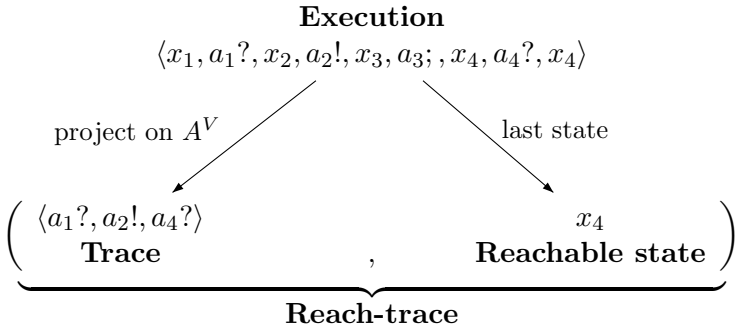


Figure 4.3: Illustration of the connection between executions, (reach-)traces, and reachable states.

For the final, infinite, execution we find the trace

$$\langle repair, repair, repair, \ldots\rangle.$$

We have that every state of P_{RC} is reachable. The explicit-divergence state $\perp$ is reachable by definition.

Figure 4.3 depicts the relationship between execution, trace, reachable state, and reach-trace.

4.4 Fairness

We now discuss how to restrict the set of executions that are to be considered for an IOA. The main reason to do so is to exclude unrealistic executions and consider only *fair* executions.

The question what constitutes an unrealistic execution depends on the assumptions made on the systems that are being modelled. Indeed, very many notions of fairness have been suggested in the modelling of generative and reactive systems. In this thesis we make somewhat different fairness assumptions than originally used for IOA, because we aim to use IOA in the larger context of compositional Markov models.

The maximal progress assumption. Consider the IOA P_1 in Figure 4.4. The ND-executions of this IOA are x, xcy, and $xcydz$. We see that the first two executions stop in an unstable state where the transitions $x \xrightarrow{c} y$ respectively $y \xrightarrow{d} z$ are enabled. We might ask if it is reasonable for an IOA execution to simply decide to stop even if one or more locally-controlled transitions are enabled. We consider such executions to be *unfair* because we make the *maximal progress assumption* [23].

Whenever any transition is enabled, the execution of the next transition may not be delayed.

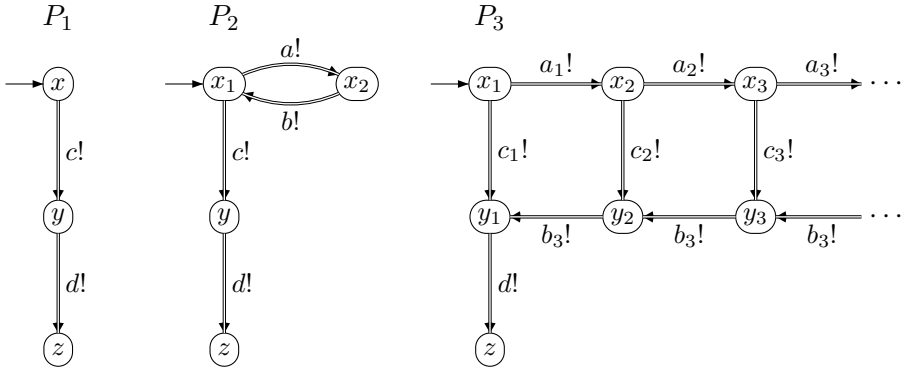


Figure 4.4: A selection of IOAs that illustrate the different notions of fairness.

That is, no execution ends in a state with enabled (locally-controlled) transitions. Note that the maximal progress assumption does not tell us *which* enabled transition should be executed, but it tells us *some* transition must be executed immediately. This may also be an input transition. The main consequence of the maximal progress assumption is that every finite fair execution ends in a stable state or in the explicit divergence state $\perp$.

The strong transition-fairness assumption. Now, let's look at the IOA P_2 in Figure 4.4. The following executions of P_2 all satisfy the maximal progress assumption.

$$\{x_1(ax_2bx_1)^i cydz \mid i \in \mathbb{N}\} \cup \{x_1(ax_2bx_1)^\omega\}.$$

For the infinite execution $x_1(ax_2bx_1)^\omega$ we see that the state x_1 is visited infinitely often and the transition (x_1, c, y) is enabled infinitely often. We call such an execution unfair, as it ignores this transition infinitely often. We make the following strong transition-fairness assumption.

If any set of locally-controlled transitions is enabled infinitely often in an execution, then transitions from the set appear infinitely often in that execution.

This is a strong fairness assumption since the requirement for fairness only supposes that the set of transitions is enabled infinitely often, not that it is enabled almost always (as is the case for weak fairness assumptions such as the one used by Lynch and Tuttle for IOA [33]). Note also that we reason about sets of transitions and not about actions or sets of actions (as is the case for the fairness assumption used by Lynch and Tuttle [33]). Consider IOA P_2 where all actions are identical (i.e., $a = b = c = d$). For this IOA the execution $x_1(ax_2ax_1)^\omega$ would still violate our fairness assumption, since although the action a appears infinitely often in the execution, the *transition* (x_1, a, y) does not. Finally, IOA P_3 shows the need to reason about sets of transitions and not single transitions. For this IOA, the infinite execution $x_1a_1x_2a_2x_3a_3x_4a_4\ldots$ is unfair since the set of transitions $\{(x_i, c_i, y_1)\}$ is enabled infinitely often, but never appears in the execution.

One of the reason for this very strong fairness assumption is the fact that we will use weak bisimulation (see Subsection 4.7.3) to equate IOA. Consider the IOA P_1, P_2, and P_3 where the actions a, b, c respectively a_i, b_i, c_i for all $i \in \mathbb{N}$ are internal instead of output actions. In this case we find that these three IOA are all weakly bisimilar. We also see that each IOA has the same set of fair reach-traces, namely $\{(d, z), (d, \perp), (\epsilon, \perp)\}$. This is crucial, because we want our notion of weak bisimulation to preserve the fair reach-traces of IOA. It is no surprise then that our notion of fairness is very close to the notion of fairness incorporated in the axiomatisation of weak bisimulation for IMCs [23], since weak bisimulation for I/O-IMCs, which we also apply to IOA, is very close to weak bisimulation for IMCs. The main difference is that for I/O-IMCs the fairness assumption is applied to sets of output and internal (i.e., locally-controlled) transitions, while the fairness assumption for IMCs is applied only to internal transitions.

The explicit divergence assumption. From the definition of a divergent state, it is clear that, whenever an execution visits a divergent state, all subsequent states are divergent. This means that, the moment one divergent state is visited, (local) time divergence is inescapable and the IOA must perform infinitely many steps in zero time. Our last fairness assumption, the explicit divergence assumption, ensures that all occurrences of local time divergence are made explicit.

Whenever an execution visits a divergent state, the execution is divergent, i.e., it ends in the explicit-divergence state $\perp$.

Definition 29. *An execution σ of P is explicit-divergence fair (EDF) if it satisfies the following statements.*

1. *If σ is finite, $last(\sigma)$ is stable or $last(\sigma) = \perp$.*
2. *If σ is infinite, then for any set of locally-controlled transitions $R \subset R^I$ we have that if R is enabled infinitely often along σ then σ contains infinitely many transitions from R.*
3. *If σ visits any divergent state, then σ must be divergent, i.e., σ is finite and $last(\sigma) = \perp$. In this case, σ is locally divergent.*

If an execution is fair then its corresponding trace is also fair. If a finite execution is fair, then its last state is said to be fairly reachable. We write $FairEx(P)$ for the set of all fair executions of P, $FairTr(P)$ for the fair traces, $FairReach(P)$ for the fairly reachable states, and $FairRT(P)$ for the set of fair reach-traces. Similarly we write $FairEx(x)$, $FairTr(x)$, $FairReach(x)$, and $FairRT(x)$ for the fair execution fragments, fair traces, fairly reachable states, and fair reach-traces of a state $x \in S$.

From here on out we will refer to explicit-divergence fair executions simply as fair executions. Our use of explicit divergence has an important consequence for closed IOA.

Theorem 9. *All fair executions of a closed IOA are finite.*

Proof. Consider a closed IOA $P = \langle S, A, R^I, x_I \rangle$. Since $A^I = \emptyset$ we find that R^I contains only locally-controlled transitions. It follows that all stable states of P have no outgoing transitions. We now prove Theorem 9 by contradiction. Assume then that σ is an infinitely long fair execution of P. Since stable states have no outgoing transitions and divergent states may not appear in σ due to the third fairness assumption, we have that the i-th state of σ, x_i, is neither stable nor divergent. There must then be a finite path ρ_i from x_i to a stable state. It is obvious that not all transitions of ρ_i are part of σ (since ρ_i contains a stable state and σ does not). Let (y_i, a_i, z_i) be the first transition of ρ_i which is not part of σ. The set $\{(y_i, a_i, z_i) \mid i \in \mathbb{N}\}$ must be enabled infinitely often along σ. If not, then we can find an index j after which the set $\{(y_i, a_i, z_i)\}$ is never enabled again. But for the $j+1$-th state of σ we must again find a path to a stable state and a transition $(y_{j+1}, a_{j+1}, z_{j+1})$, otherwise this state would be divergent. The fact that the set $\{(y_i, a_i, z_i) \mid i \in \mathbb{N}\}$ is enabled infinitely often but never appears in σ is a contradiction with the fact that σ is fair. □

In the next section we will see that Theorem 9 means that, for practical purposes, we can restrict our attention to fair finite executions of IOA. The set of fair finite executions of an IOA P is denoted $\mathit{FinFairEx}(P)$ and it can easily be shown that it is simply the union of the set of non-divergent executions that end in a stable state and the set of all finite divergent executions.

4.5 Parallel Composition

We will now discuss a composition operation for IOA as well as its properties. This composition operation allows us to combine different IOA to form new ones.

Two IOA may communicate with each other by sending and receiving events. Such events are labelled with actions and we then have input, output, and internal events labelled with input, output, and internal actions. For instance, the IOA P_{RC} may send an event labelled *fail* by executing the transition (**failing**, *fail*, **down**). When the IOA subsequently receives an event labelled *repair* from another IOA it will execute the transition (**down**, *repair*, **recovering**). We say the *repair*-transition of P_{RC} *synchronises* with a *repair*-transition of another IOA. A central feature of IOA is that it uses *asymmetric synchronisation* in which, for every synchronisation, there is at most one IOA that controls when the synchronisation may take place. For instance, the IOA P_{RC} has no control over when events labelled *repair* occur since this input-action is always enabled.

It should be noted that *symmetric synchronisation* or hand-shake synchronisation is used widely in the modelling of generative systems [35, 27]. In symmetric synchronisation a synchronised transition can only occur if all participating components enable the transition. We can interpret asymmetric synchronisation as a special case of symmetric synchronisation in which it happens to be the case that each action is always enabled for all but one of the participating components.

To make sure synchronisation is asymmetric we restrict the composition of IOA to IOA that have disjoint locally-controlled actions. Such IOA are called *compatible*. The

result is that each action is controlled by at most one IOA.

Definition 30. *Two IOA P_1 and P_2 with input actions A_1^I respectively A_2^I, output actions A_1^O respectively A_2^O, and internal actions A_1^H respectively A_2^H, are compatible if*

1. *they do not share any output actions,*

$$A_1^O \cap A_2^O = \emptyset,$$

and

2. *their internal actions are unique,*

$$A_1^H \cap (A_2^I \cup A_2^O \cup A_2^H) = \emptyset$$

and

$$A_2^H \cap (A_1^I \cup A_1^O \cup A_1^H) = \emptyset.$$

A set of IOA is compatible if the IOA in the set are pairwise compatible.

The parallel composition of two IOA is again an IOA, whose state space is the cross-product of the state spaces of the components. Transitions that are labelled by actions in the shared alphabet of the two components are synchronised, whereas the rest of the transitions are *interleaved*.

Definition 31. *Let $P_1 = \langle S_1, A_1, R_1^I, \hat{x}_1\rangle$ and $P_2 = \langle S_2, A_2, R_2^I, \hat{x}_2\rangle$ be two compatible IOA. The* parallel composition[1] *$P_1 \| P_2$ of P_1 and P_2 is an IOA with*

- *state space $S_1 \| S_2 = \{x_1 \| x_2 \mid x_1 \in S_1, x_2 \in S_2\}$,*
- *actions $A_1 \cup A_2$, with output actions $A^O = A_1^O \cup A_2^O$, input actions $A^I = (A_1^I \cup A_2^I) - A^O$, and internal actions $A^H = A_1^H \cup A_2^H$,*
- *transitions*

$$\begin{aligned} R^I = &\{x_1 \| x_2 \xrightarrow{a} y_1 \| y_2 \mid a \in A_1 \cap A_2, (x_1, a, y_1) \in R_1^I, (x_2, a, y_2) \in R_2^I\} \\ &\cup \{x_1 \| x_2 \xrightarrow{a} y_1 \| x_2 \mid a \in A_1 - A_2, (x_1, a, y_1) \in R_1^I\} \\ &\cup \{x_1 \| x_2 \xrightarrow{a} x_1 \| y_2 \mid a \in A_2 - A_1, (x_2, a, y_2) \in R_2^I\}, \text{ and} \end{aligned}$$

- *initial state $\hat{x}_1 \| \hat{x}_2$.*

The IOA $P_1 \| P_2$ can easily be shown to be input-enabled and finitely branching.

Since the parallel composition of two IOA is another IOA parallel compositions of finitely many IOA can be achieved by composing again with a third IOA and so forth. Note that composing in parallel infinitely many IOA may lead to infinite branching and for this reason we consider only finite parallel compositions in this thesis.

[1] We recycle the parallel composition operator for states. It will always be clear from the context which of the two operators is meant.

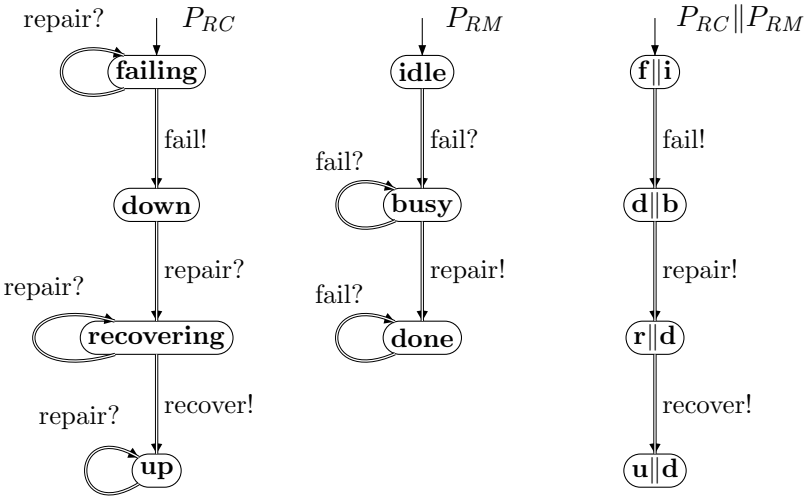


Figure 4.5: Example of the parallel composition of two compatible IOA. In the parallel composition, the names of the component states are abbreviated.

Proposition 3. *Parallel composition for IOA is associative up to isomorphism. Given three pair-wise compatible IOA P_1, P_2, and P_3 we have*

- $(P_1 \| P_2) \| P_3 \equiv P_1 \| (P_2 \| P_3)$, *and*
- *if P_1 and P_2 are compatible with P_3 then $P_1 \| P_2$ is compatible with P_3.*

Proof. Standard. □

The associativity of parallel composition means that the order in which the IOA are composed makes no difference semantically[2]. For this reason we will leave out the brackets when parallel composing more than two IOA. That is, we write $P_1 \| P_2 \| P_3$ instead of $(P_1 \| P_2) \| P_3$ or $P_1 (\| P_2 \| P_3)$.

Example 14. *As an example, Figure 4.5 shows the compatible IOA P_{RC} (from Example 10), P_{RM} (which models a generic repair man), and their parallel composition $P_{RC} \| P_{RM}$.*

The inverse of parallel composition is called *projection*. Any state, execution, trace, or reach-trace of a composite IOA $P_1 \| P_2$ can be projected back onto one of its components.

Definition 32. *Given two compatible IOA P_1 and P_2, we define the following projections from $P_1 \| P_2$ onto P_1.*

- *The projection of state $x \| y$ of $P_1 \| P_2$ onto P_1, denoted $x \| y \downarrow P_1$ is x and the projection of the distinguished state $\bot$ onto P_1, denoted $\bot \downarrow P_1$ is again the state $\bot$.*

[2]We will see in Chapter 9 that the order of composition may make a very important difference in practice.

- *The projection of an execution σ of $P_1 \| P_2$ onto P_1, denoted $\sigma \downarrow P_1$ is defined recursively.*

$$\sigma \downarrow P_1 = \begin{cases} \langle x_1 \rangle, & \text{if } \sigma = \langle x_1 \| x_2 \rangle, \\ \langle x_1, \bot \rangle, & \text{if } \sigma = \langle x_1 \| x_2, \bot \rangle, \\ \sigma' \downarrow P_1, & \text{if } \sigma = \langle x_1 \| x_2, a \rangle \circ \sigma', a \notin A_1, \\ \langle x_1, a \rangle \circ \sigma' \downarrow P_1, & \text{if } \sigma = \langle x_1 \| x_2, a \rangle \circ \sigma', a \in A_1. \end{cases}$$

- *The projection of a trace w of $P_1 \| P_2$ onto P_1, denoted $w \downarrow P_1$ is $w \downarrow A_1^V$, where A_1^V are the visible actions of P_1.*

- *The projection of a reach-trace $(w, x \| y)$ of $P_1 \| P_2$ onto P_1, denoted $(w, x \| y) \downarrow P_1$ is $(w \downarrow P_1, x \| y \downarrow P_1)$.*

Projections from $P_1 \| P_2$ onto P_2 are defined symmetrically. All projections use the same notation, but it will always be clear from context which projection is meant.

4.5.1 Modularity results

In the following we consider two compatible IOA $P_1 = \langle S_1, A_1, R_1^I, \hat{x}_1 \rangle$ and $P_2 = \langle S_2, A_2, R_2^I, \hat{x}_2 \rangle$. A crucial consequence of asymmetric synchronisation is that the enabledness of actions is preserved by parallel composition and projection.

Proposition 4. *For any two states $x_1 \in S_1$, $x_2 \in S_2$ and any action $a \in A_1 \cup A_2$ we have*

$$a \in En(x_1 \| x_2) \Leftrightarrow a \in En(x_1) \vee a \in En(x_2).$$

Proof. This is a direct consequence of the input-enabledness of IOA and the compatibility of P_1 and P_2. □

We can restate Proposition 4 in terms of enabled sets. We have

$$\begin{aligned} En(x_1 \| x_2) &= En(x_1) \cup En(x_2), \\ En(x_1) &= En(x_1 \| x_2) \downarrow A_1, \text{ and} \\ En(x_2) &= En(x_1 \| x_2) \downarrow A_2. \end{aligned}$$

Proposition 4 has as a consequence that stability of states is also preserved by composition and projection.

Corollary 4. *For any pair of states $x_1 \in S_1$, $x_2 \in S_2$ we have*

$$st(x_1 \| x_2) \Leftrightarrow st(x_1) \wedge st(x_2). \tag{4.1}$$

and equivalently

$$\neg st(x_1 \| x_2) \Leftrightarrow \neg st(x_1) \vee \neg st(x_2). \tag{4.2}$$

We will now study the relationship between executions of P_1 and P_2 and executions of their parallel composition. To do this, we extend the notion of compatibility to executions of IOA. Two executions of IOA P_1 and P_2 are compatible if they "agree" on the order in which actions from the shared alphabet occurs.

Definition 33. *The executions $\sigma_1 \in Ex(P_1)$ and $\sigma_2 \in Ex(P_2)$ are compatible if*

$$\sigma_1 \downarrow (A_1 \cap A_2) = \sigma_2 \downarrow (A_1 \cap A_2)$$

and either both executions are non-divergent or both executions are explicitly divergent. The traces $w_1 \in Tr(P_1)$ and $w_2 \in Tr(P_2)$ are compatible if

$$w_1 \downarrow (A_1 \cap A_2) = w_2 \downarrow (A_1 \cap A_2).$$

We define compatibility of execution fragments and their associated traces in the same way.

Our first result is that executions are preserved by projecting and, furthermore, the resulting projected executions are compatible.

Proposition 5. *For any execution σ_3 of $P_1 \| P_2$ we have that $\sigma_3 \downarrow P_1$ is an execution of P_1 and $\sigma_3 \downarrow P_2$ is an execution of P_2 and these two executions are compatible.*

Proof. We first prove that Proposition 5 holds in the case that σ_3 is a finite, possibly divergent, *execution fragment* (i.e., σ_3 need not start in the initial state) by induction on the length n of σ_3. For $n = 0$ we have that the projections are empty and the proposition trivially holds. We now assume the proposition holds for paths of length n and show that it follows that the proposition holds for paths of length $n + 1$. We have $\sigma_3 = \langle x_1 \| x_2, a \rangle \circ \sigma_3'$ for some path σ_3' of length n, action $a \in A_1 \cup A_2$, and states $x_1 \in S_1$, $x_2 \in S_2$. We now have

$$\sigma_3 \downarrow P_1 = \begin{cases} \sigma_3' \downarrow P_1, & \text{if } a \notin A_1, \\ \langle x_1, a \rangle \circ \sigma_3' \downarrow P_1, & \text{if } a \in A_1. \end{cases}$$

For the projection of $\sigma_3 \downarrow P_1$ onto $A_1 \cap A_2$ we then find

$$(\sigma_3 \downarrow P_1) \downarrow (A_1 \cap A_2) = \begin{cases} (\sigma_3' \downarrow P_1) \downarrow (A_1 \cap A_2), & \text{if } a \notin A_1 \cap A_2 \\ \langle a \rangle \circ (\sigma_3' \downarrow P_1) \downarrow (A_1 \cap A_2), & \text{if } a \in A_1 \cap A_2. \end{cases}$$

Similarly, we find for the projection of σ_3 onto P_1 that

$$(\sigma_3 \downarrow P_2) \downarrow (A_1 \cap A_2) = \begin{cases} (\sigma_3' \downarrow P_2) \downarrow (A_1 \cap A_2), & \text{if } a \notin A_1 \cap A_2 \\ \langle a \rangle \circ (\sigma_3' \downarrow P_2) \downarrow (A_1 \cap A_2), & \text{if } a \in A_1 \cap A_2. \end{cases}$$

By the induction assumption it follows that $(\sigma_3' \downarrow P_1) \downarrow (A_1 \cap A_2) = (\sigma_3' \downarrow P_2) \downarrow (A_1 \cap A_2)$ and then $\sigma_3 \downarrow P_1$ must be compatible with $\sigma_3 \downarrow P_2$.

For an infinite execution σ_3 we prove the proposition by contradiction. Assume the two projections of σ_3 onto P_1 and P_2 are not compatible. Then there exists a finite prefix σ_3' of σ_3 for which we have $(\sigma_3' \downarrow P_1) \downarrow (A_1 \cap A_2) \neq (\sigma_3' \downarrow P_2) \downarrow (A_1 \cap A_2)$. But this is a contradiction with the fact that the proposition holds for finite paths. □

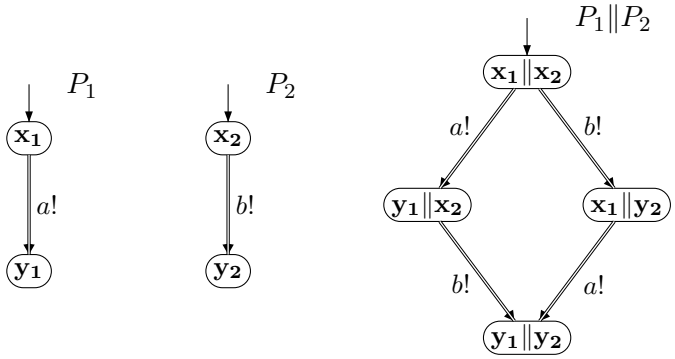


Figure 4.6: Example of the interleaving of independent actions.

Corollary 5. *For any trace w of $P_1 \| P_2$, any reachable state $x \| y$ of $P_1 \| P_2$, and any reach-trace $(w', x' \| y')$ of $P_1 \| P_2$ we have that $w \downarrow P_1$ and $w \downarrow P_2$ are compatible traces of P_1 and P_2, respectively, x and y are reachable states for P_1 and P_2, respectively, and $(w', x' \| y') \downarrow P_1$ and $(w', x' \| y') \downarrow P_2$ are reach-traces of P_1 and P_2, respectively.*

It would be natural to try to extend the notion of parallel composition to executions. Given two compatible IOA P_1 and P_2, the parallel composition of compatible executions $\sigma_1 \in Ex(P_1)$ and $\sigma_2 \in Ex(P_2)$ would then yield an execution of $P_1 \| P_2$. However, we will see that this is impossible, because the executions of $P_1 \| P_2$ generally hold more information than the associated executions in P_1 and P_2.

Example 15. *Consider the two compatible IOA P_1 and P_2 in Figure 4.6. We can see that the only fair, non-divergent execution of P_1 is $\sigma_1 = \langle x_1, a, y_1 \rangle$ and similarly the only fair, non-divergent execution of P_2 is $\sigma_2 = \langle x_2, b, y_2 \rangle$. These two executions are trivially compatible since the shared alphabet of P_1 and P_2 is empty. We could now ask what the parallel composition of σ_1 and σ_2 would look like, but if we inspect the IOA $P_1 \| P_2$ we see that there are in fact two possibilities. Either the transition (x_1, a, y_1) occurs before the transition (x_2, b, y_2) and the combined execution is $\langle x_1 \| x_2, a, y_1 \| x_2, b, y_1 \| y_2 \rangle$, or the transitions occur in the reverse order and the combined execution is $\langle x_1 \| x_2, b, x_1 \| y_2, a, y_1 \| y_2 \rangle$.*

The reason that component executions do not completely determine an execution is that we use an *interleaving semantics*. This means that, whenever two transitions are enabled *one will occur before the other*. That is, the execution of transitions is totally ordered. The order in which enabled transitions from different components occur is exactly the extra information contained in the executions of a composed IOA compared to the executions of its components.

However, Lynch and Tuttle have shown a slightly weaker result. Given two compatible executions of P_1 and P_2, there must exist some execution of $P_1 \| P_2$ which projects back onto the executions of P_1 and P_2.

Theorem 10 ([33])**.** *Given two compatible executions σ_1 and σ_2 of P_1 respectively P_2 there exists an execution σ_3 of $P_1 \| P_2$ such that $\sigma_3 \downarrow P_1 = \sigma_1$ and $\sigma_3 \downarrow P_2 = \sigma_2$.*

Corollary 6. *Given compatible traces w_1 and w_2 of P_1 respectively P_2, and compatible reach-traces (w_1, x) and (w_2, y) of P_1 respectively P_2, there exists a trace w and a reach-trace $(w, x\|y)$ of $P_1\|P_2$ such that $w{\downarrow}P_1 = w_1$, $w{\downarrow}P_2 = w_2$, $(w, x\|y){\downarrow}P_1 = (w_1, x)$, and $(w, x\|y){\downarrow}P_2 = (w_2, y)$.*

4.5.2 Composition and fairness

It turns out the modularity results we established for executions, traces, and reachable states also hold for their fair counterparts.

Theorem 11. *Let P_1 and P_2 be two compatible IOA. For any fair execution of σ_3 of $P_1\|P_2$ we have that $\sigma_3{\downarrow}P_1$ is a fair execution of P_1 and $\sigma_3{\downarrow}P_2$ is a fair execution of P_2 and these two executions are compatible.*

Proof. We prove by contradiction that the projected executions are indeed fair. We assume then that the execution σ_3 of $P_1\|P_2$ is fair, but execution $\sigma_1 = \sigma_3{\downarrow}P_1$ is not. It follows that one of the three fairness conditions does not hold for σ_1. We consider each condition separately.

- Consider the case that σ_1 is finite and its last state $x_1 = \mathit{last}(\sigma_1)$ is unstable. If σ_3 is also finite, then its last state must be of the form $x_1\|x_2$ for some state $x_2 \in S_2$. Corollary 4.1 then gives us that this last state of σ_3 is also unstable, which is a contradiction with the fact that σ_3 is fair. For the case that σ_3 is infinite, we have that σ_3 must visit infinitely many states in $\{x_1\} \times S_2$. From the instability of x_1 it follows that there is a transition (x_1, a, y_1) enabled in x_1. Proposition 4 then gives us that there is a set of transitions $\{(x_1\|z_i, a, y_1\|z_i') \mid z_i, z_i' \in S_2\}$ in $P_1\|P_2$, that is enabled infinitely often for σ_3. The fairness of σ_3 then proscribes that this transition is also executed infinitely often, but this means the transition (x_1, a, y_1) is also executed infinitely often by σ_1, which is a contradiction with the fact that σ_1 is finite.

- Consider the case that σ_1 is infinite and there is some set of actions $R \subset R_1^I$ that is enabled infinitely often along σ_1 but which is not executed infinitely often by σ_1. It follows immediately from Proposition 4 and Corollary 4 that the same then holds for σ_3 for the set

$$R' = \{(x_1\|x_2, a, y_1\|y_2) \mid (x_1, a, y_1) \in R, x_2, y_2 \in S_2\} \cap R_3^I.$$

  This is a contradiction with the fact that σ_3 is fair.

- Consider the case that σ_1 visits a divergent state but does not end in the explicit-divergence state $\perp$. From the definition of projection it follows that σ_3 also does not end in the explicit-divergence state $\perp$. However, Corollary 4.3 gives us that σ_3 visits a divergent state, which contradicts the fairness of σ_3.

□

This result also extends to fair traces, reachable states, and reach-traces.

Corollary 7. *For any fair trace w of $P_1 \| P_2$, any fairly reachable state $x \| y$ of $P_1 \| P_2$, and any fair reach-trace $(w', x' \| y')$ of $P_1 \| P_2$ we have that $w \downarrow P_1$ and $w \downarrow P_2$ are compatible fair traces of P_1 respectively P_2, x and y are fairly reachable states for P_1 respectively P_2, and $(w', x' \| y') \downarrow P_1$ and $(w', x' \| y') \downarrow P_2$ are fair reach-traces of P_1 respectively P_2.*

We have seen that we cannot construct an execution of $P_1 \| P_2$ from compatible executions of P_1 and P_2. However, we can easily see that given compatible executions σ_1 and σ_2 of P_1 respectively P_2 there must exists an execution σ_3 of $P_1 \| P_2$ such that the projection of σ_3 onto P_1 is σ_1 and the projection of σ_3 onto P_2 is σ_2 [33]. Additionally, if σ_1 and σ_2 are *finite* fair executions, then there exists an execution σ_3 as above that is also finite and fair.

Theorem 12. *Let P_1 and P_2 be two compatible IOA and let σ_3 be a finite sequence of states of $P_1 \| P_2$ interleaved with actions of $P_1 \| P_2$, i.e.,*

$$\sigma_3 = \langle x_0 \| y_0, a_0, x_1 \| y_1, \ldots, a_{n-1}, x_n \| y_n \rangle,$$

for some states $x_0, \ldots, x_n \in S_1$, $y_0, \ldots, y_n \in S_2$, and $a_0, \ldots, a_{n-1} \in A_1 \cup A_2$. We have that, σ_3 is a finite fair execution of P_3, if and only if for any index $0 \leq i < n$ it holds that

$$a_i \notin A_1 \implies x_i = x_{i+1} \tag{4.3}$$

and

$$a_i \notin A_2 \implies y_i = y_{i+1} \tag{4.4}$$

and the projections of σ_3 as per Definition 32, $\sigma_1 = \sigma_3 \downarrow P_1$ and $\sigma_2 = \sigma_3 \downarrow P_2$, are compatible, finite, *and* fair *executions of P_1 and P_2, respectively. Moreover, the same holds for $\sigma_3 \circ \langle \perp \rangle$.*

Proof. The "if" of Theorem 12 follows directly from Theorem 11.

To prove the "only if", we first show that σ_3 is a finite execution of $P_1 \| P_2$. Given conditions (4.3) and (4.4) it is easy to show that σ_1 starts with state x_0 and σ_2 starts with state y_0. It then remains to show that for any index $0 \leq i < n$ we have that the triple $\langle x_i \| y_i, a_i, x_{i+1} \| y_{i+1} \rangle$ is a transition of $P_1 \| P_2$. For the case $a_i \in A_1 \cap A_2$ we have that the projections σ_1 and σ_2 will contain the transitions $\langle x_i, a_i, x_{i+1} \rangle$ respectively $\langle y_i, a_i, y_{i+1} \rangle$. It then follows from the definition of parallel composition that $P_1 \| P_2$ indeed contains the transition in question. For the case $a_i \in A_1$, $a_i \notin A_2$ we have that the projection σ_1 will contain the transition $\langle x_i, a_i, x_{i+1} \rangle$ and $y_i = y_{i=1}$, because of (4.4). Again it follows from the definition of parallel composition that the transition in questions appears $P_1 \| P_2$. The remaining case $a_i \notin A_1$, $a_i \in A_2$ proceeds in a similar way. Adding $\perp$ to the end of σ_3 does not affect the fact that σ_3 is a finite execution of $P_1 \| P_2$.

We now show by contradiction that σ_3 is fair. Assume then that σ_3 violates one of the three fairness conditions. For the case that σ_3 ends in an unstable state $x_n \| y_n$ we have, by (4.2), that either x_n is unstable or y_n is unstable and then also one of the projections of σ_3 is unfair which is a contradiction. The second fairness condition cannot be violated by σ_3 since it is finite. Finally, assume that σ_3 visits a divergent state but does not end in the explicit divergence state $\perp$. Since σ_3 is an execution of $P_1 \| P_2$ it then

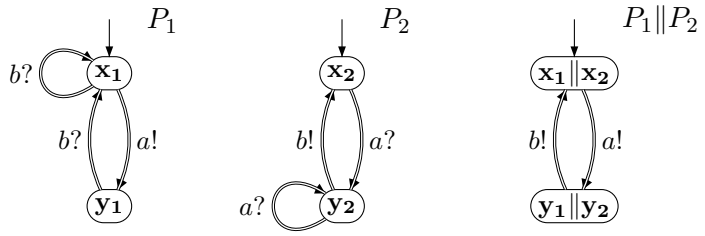


Figure 4.7: Two IOA with infinite fair executions and their parallel composition.

follows that the last state of σ_3 is unstable (since no stable state can be reached from a divergent state). But we have already seen that this leads to a contradiction with the assumption that σ_1 and σ_2 are fair. □

We might ask why Theorem 12 restricts to *finite* executions. We will give an example that shows why we need this restriction. Consider the compatible IOA P_1 and P_2 depicted in Figure 4.7. The infinite executions $\sigma_1 = (x_1, a, y_1, b)^\omega$ of P_1 and $\sigma_2 = (x_2, a, y_2, b)^\omega$ are infinite, fair, and compatible. Note especially that the states y_1 and x_2 are stable. If we now investigate the parallel composition of P_1 and P_2 we find that $\sigma_3 = (x_1 \| x_2, a, y_1 \| y_2, b)^\omega)$ is the only execution of σ_3 such that $\sigma_3 \downarrow P_1 = \sigma_1$ and $\sigma_3 \downarrow P_2 = \sigma_2$. However, we have that both $x_1 \| x_2$ and $y_1 \| y_2$ are unstable and then both these states are also divergent. It then follows that σ_3 is not a fair execution as it does not end in the explicit-divergence state $\perp$. Note that there do not exist two non-divergent, finite, fair, and compatible executions of P_1 and P_2, since one of the executions will end in an unstable state. This matches the fact that $P_1 \| P_2$ has no fair non-divergent executions.

The above counter-example is not so surprising in light of Theorem 9, which states that all fair executions of a closed IOA are finite. Since the same does not hold for IOA that are not closed we see that fairness of infinite executions cannot be preserved by parallel composition. It is important to note that this is a consequence of our use of explicit divergence. In the original treatment of IOA, explicit divergence is not used and then infinite executions are indeed preserved by parallel composition [33].

Corollary 8. *Let P_1 and P_2 be two compatible IOA such that their parallel composition $P_1 \| P_2$ is closed. For an infinite fair execution σ_1 of P_1 and a fair execution σ_2 of P_2 we have that any execution σ_3 of $P_1 \| P_2$ such that $\sigma_3 \downarrow P_1 = \sigma_1$ and $\sigma_3 \downarrow P_2 = \sigma_2$ is unfair.*

Proof. From the definition of projection it follows that σ_3 must be infinite. Corollary 8 then follows directly from Theorem 9. □

Again we have that the results extend to fair traces and reach-traces.

Corollary 9. *Given a finite sequence of visible actions of $P_1 \| P_2$, $w = \langle a_1, \ldots, a_n \rangle$ and a state $x \| y$ of $P_1 \| P_2$, we have that,*

1. *$w \downarrow P_1$ and $w \downarrow P_2$ are finite fair traces of P_1 and P_2, respectively, if and only if w is a finite fair trace of $P_1 \| P_2$,*
2. *if $x \| y$ is fairly reachable in $P_1 \| P_2$ then x and y are fairly reachable in P_1 and P_2, respectively, and*
3. *$\langle w \downarrow P_1, x \rangle$ and $\langle w \downarrow P_2, y \rangle$ are fair reach-traces of P_1 and P_2, respectively if and only if $\langle w, x \| y \rangle$ is a fair reach-trace of $P_1 \| P_2$.*

Proof. Recall that the fact that $\langle w \downarrow P_1, x \rangle$ and $\langle w \downarrow P_2, y \rangle$ are fair reach-traces of P_1 respectively P_2 means that there must exist fair executions σ_1 and σ_2 that have traces $w \downarrow P_1$ respectively $w \downarrow P_2$ and end in states x respectively y. Furthermore, these executions must be compatible because their traces are both projections of w. Under these considerations we see that the corollary follows from Theorems 11 and 12. □

Crucially, fair reachability is not preserved by parallel composition (i.e., the reverse of the second statement in Corollary 9 does not hold). This means that the reachability of states in an IOA depends on its environment. Specifically, it depends on the traces of its environment, which is why the third statement of Corollary 9 holds in both directions. It is important to note that *unreachability* is preserved by parallel composition. If a state x is not reachable in P_1 then no state $x \| y$ will be reachable in $P_1 \| P_2$.

4.6 Hiding

We can make IOA models more abstract by *hiding* certain actions.

Definition 34. *Given an IOA $P = \langle S, A, R^I, \hat{x} \rangle$ and a subset of its output actions $B \subseteq A^O$,* hiding *B in P yields the IOA $P \backslash B = \langle S, A_2, R^I, \hat{x} \rangle$ where we have $A_2^O = A^O - B$ and $A_2^H = A^H \cup B$.*

Hiding simply changes the role of output actions to internal actions. Note that hiding actions in an IOA may make it incompatible to other IOA. That is, if two IOA P_1 and P_2 with actions A_1 respectively A_2 are compatible, then $P_1 \backslash B$ is compatible with P_2 if and only if $B \cap A_2 = \emptyset$. In general, actions are hidden after they are used in a parallel composition to synchronise different component IOA.

It is easy to see that any execution of P is also an execution of $P \backslash B$. Since the definition of fairness for IOA executions does not distinguish between output and internal actions (both are locally-controlled), hiding actions in an IOA also does not change the set of fair executions.

Proposition 6. *Given an IOA $P = \langle S, A, R^I, \hat{x} \rangle$ and a subset of its output actions $B \subseteq A^O$, a sequence of states in S interleaved with actions in A is a (fair) execution of P if and only if it is a (fair) execution of $P \backslash B$*

Proof. Trivial. □

However, hiding does change the set of fair finite traces and fair reach-traces. This is caused by the fact that traces only list visible actions, and the set of visible actions is changed by hiding. In the following, we extend hiding to traces in an intuitive way. A trace $w\backslash B$ is the trace w where all actions in B have been removed.

Proposition 7. *Given an IOA $P = \langle S, A, R^I, \hat{x}\rangle$, a subset of its output actions $B \subseteq A^O$, a finite sequence $\bar{w}$ of actions from $A^V\backslash B$, and a state $x \in S_\perp$, we have*

1. *$\bar{w}$ is a fair finite trace of $P\backslash B$ if and only if there exists a fair finite trace w of P such that $w\backslash B = \bar{w}$,*
2. *$(\bar{w}, x)$ is a fair reach-trace of $P\backslash B$ if and only if there exists a reach-trace (w, x) of P such that $w\backslash B = \bar{w}$, and*
3. *x is fairly reachable in $P\backslash B$ if and only if x is fairly reachable in P.*

The same holds for the general finite traces, reach-traces and reachable states (i.e., without fairness).

Proof. Trivial. □

Note, that Proposition 7 does not allow us to determine the (fair) traces of P from the (fair) traces of $P\backslash B$. However, the (fairly) reachable states of P are preserved by hiding.

4.7 Equivalences

We have already seen in Chapter 3 that equivalence relations are an important concept, and we now discuss some important equivalence relations in the context of IOA. Our main focus is on reachability properties of IOA, because this links to the considerations in the preceding and subsequent chapters.

In essence, we want to identify an equivalence relation that equates IOA with the same reachability properties. However, given two IOA P_1 and P_2 with disjoint state spaces S_1 and S_2, we must have a way of equating states of P_1 and P_2 in order to meaningfully compare their reachability properties. Recall from Section 2.1 that we have assumed that there is a congruence relation $=_s$ on the set of all states which tells us which states can be distinguished.

We will only compare IOA which are *comparable*, in the sense that they have similar action signatures and, for practical reasons, disjoint state spaces.

Definition 35. *Two IOA P_1 and P_2 are* comparable *when*

- *they have the same output actions,*
- *they have the same input actions,*

- *an internal action of P_1 is not a visible action of P_2 and vice versa, and*
- *their state spaces are disjoint.*

With respect to the restriction that the state spaces are disjoint, recall that for any IOA P and any subset $S \subset S_{\mathsf{all}}$ we can find an isomorphic IOA P' up to $=_s$ such that its state space is disjoint from S.

To compare two IOA it will be useful to combine their state spaces and transition relations. This is purely a technicality which makes it easier to define equivalence relations for IOA.

Definition 36. *Given two IOA $P_1 = \langle S_1, A_1, R_1^I, \hat{x}_1\rangle$ and $P_2 = \langle S_2, A_2, R_2^I, \hat{x}_2\rangle$, which are comparable, have disjoint state spaces S_1 and S_2, identical output actions A_1^O and A_2^O, and identical input actions A_1^I and A_2^I, the* disjoint union *of P_1 and P_2 is the IOA $P_1 \cup P_2 = \langle S_1 \cup S_1, A_1 \cup A_2, R_1^I \cup R_2^I, \hat{x}\rangle$.*

We pick the initial state of the disjoint union arbitrarily as it will not play a significant role in the remainder of this section.

4.7.1 Reachability equivalence

We can now define *reachability equivalence*. Two IOA are reachability equivalent if the sets of states they can *fairly* reach are the same with respect to $=_s$. We ignore the explicit-divergence state $\perp$, since it is fairly reachable for all IOA.

Definition 37. *Given an IOA $P = \langle S, A, R^I, \hat{x}\rangle$, an equivalence relation $\mathcal{E}$ on S is a* reachability *equivalence if for all pairs of states $x\mathcal{E}y$ we have,*

$$x' \in \mathit{FairReach}(x) \setminus \{\perp\} \implies \exists y' \in \mathit{FairReach}(y) \setminus \{\perp\} \cdot x' =_s y'.$$

We say two states $x, y \in S$ are reachability equivalent *in P, denoted $x =_r^P y$, if there exists a reachability equivalence $\mathcal{E}$ such that $x\mathcal{E}y$. We leave out the superscript when clear from context.*

Two comparable IOA P_1 and P_2 are reachability equivalent, denoted $P_1 =_r P_2$ if their initial states are reachability equivalent in the disjoint union of P_1 and P_2. I.e, we have $\hat{x}_1 =_r^{P_1 \cup P_2} \hat{x}_2$.

Reachability equivalence obviously preserves the reachability properties of IOA in isolation. It is also good to note that hiding actions does not affect the reachability of states and we then find that reachability equivalence is substitutive with respect to hiding.

Proposition 8. *Given two IOA P_1 and P_2 such that $A_1^O = A_2^O$, then we have for any set $B \subset A_1^O$*

$$P_1 =_r P_2 \implies P_1 \backslash B =_r P_2 \backslash B.$$

Proof. This is a direct consequence of the fact that our notion of fairness does not distinguish between internal and output actions. □

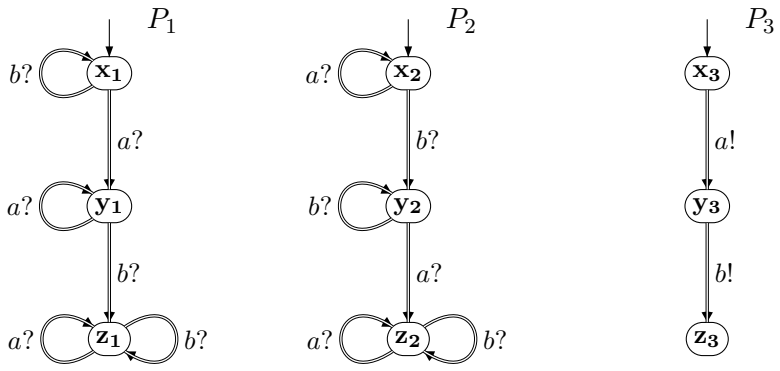


Figure 4.8: Three IOA.

4.7.2 Reach-trace equivalence

Unfortunately, reachability equivalence is not substitutive with respect to parallel composition. This means that IOA that are reachability equivalent may not be reachability equivalent after composing them in parallel with the same compatible IOA.

Example 16. *Consider the IOA P_1, P_2, and P_3 in Figure 4.8. Assume that we have $x_1 =_s x_2$, $y_1 =_s y_2$, $z_1 =_s z_2$, and all other pairs of states are different with respect to $=_s$. It is then easy to see that P_1 and P_2 are reachability equivalent. However, $P_1 \| P_3$ can reach states $x_1 \| x_3$, $y_1 \| y_3$, and $z_1 \| z_3$ whereas $P_2 \| P_3$ can reach $x_2 \| x_3$, $x_2 \| y_3$, and $y_2 \| z_3$. It is clear that $P_1 \| P_3$ is not reachability equivalent to $P_2 \| P_3$. For instance, the state $z_1 \| z_3$ is not equivalent to any of the reachable states of $P_2 \| P_3$.*

The fact that reachability equivalence is not substitutive with parallel composition is not surprising, given the fact that the executions of a parallel composition are determined by all pairs of *compatible* executions of its components. The compatibility of executions in turn depends on the *traces* of these executions. To find an equivalence relation that is substitutive with parallel composition, we must then take these traces into account.

Definition 38. *Given an IOA $P = \langle S, A, R^I, \hat{x} \rangle$, an equivalence relation $\mathcal{E}$ on S is a* reach-trace *equivalence if for all pairs of states $x \mathcal{E} y$ we have,*

$$(w, x') \in \mathit{FairRT}(x) \implies \exists (w, y') \in \mathit{FairRT}(y) \cdot x' = y' = \bot \vee x' =_s y'.$$

We say two states $x, y \in S$ are reach-trace equivalent *in P, denoted $x =_{rt}^P y$ if there exists a reach-trace equivalence $\mathcal{E}$ such that $x \mathcal{E} y$. We leave out the superscript when clear from context.*

Two comparable IOA P_1 and P_2 are reach-trace equivalent, denoted $P_1 =_{rt} P_2$ if their initial states are reach-trace equivalent in the disjoint union of P_1 and P_2.

Reach-trace equivalence is obviously coarser than reachability equivalence. Moreover it is substitutive with respect to hiding and parallel composition.

Theorem 13. *Given two IOA P_1, P_2 such that $A_1^O = A_2^O$, if P_1 is reach-trace equivalent to P_2 then*

1. *P_1 is reachability equivalent to P_2,*
2. *for any set of actions $B \subset A_1^O$ we have that $P_1 \backslash B$ is reach-trace equivalent to $P_2 \backslash B$, and*
3. *for any IOA P_3, which is compatible with both P_1 and P_2 we have that $P_1 \| P_3$ is reach-trace equivalent to $P_2 \| P_3$.*

Proof. The first two statements of Theorem 13 trivially hold. For the third statement we must show that for every fair reach-trace of $P_1 \| P_3$ there is a corresponding reach-trace of $P_2 \| P_3$. Let $\langle w, x_1 \| x_3 \rangle$ be a fair reach-trace of $P_1 \| P_3$. We then find a finite fair execution σ of $P_1 \| P_3$ such that $Tr(\sigma) = w$ and $last(\sigma) = x_1 \| x_3$. By Theorem 11 we have that $\sigma_1 = \sigma \downarrow P_1$ and $\sigma_3 = \sigma \downarrow P_3$ are compatible finite fair executions of P_1 respectively P_3. Obviously we have $last(\sigma_1) = x_1$ and $last(\sigma_3) = x_3$. Let w_1 and w_3 be the traces of σ_1 and σ_3. We then have that $\langle w_1, x_1 \rangle$ is a fair reach-trace of P_1.

Now, since P_1 is reach-trace equivalent to P_2 we find a state x_2 and an execution σ_2 of P_2 such that $Tr(\sigma_2) = w_1$ and $last(\sigma_2) = x_2 =_s x_1$. Theorem 12 allows us to combine executions σ_2 and σ_3 to find a finite fair execution σ' of $P_2 \| P_3$ such that $\sigma' \downarrow P_2 = \sigma_2$ and $\sigma' \downarrow P_3 = \sigma_3$. For the final state of σ' we have $last(\sigma') = x_2 \| x_3 =_s x_1 \| x_3$. Let w' be the trace of σ'. We then find $w' \downarrow P_2 = w_1$ and $w' \downarrow P_3 = w_3$. The order in which the actions in the shared alphabet of P_1 and P_3 (equivalently, the shared alphabet of P_2 and P_3) are then the same for both w and w'. It remains to be shown that the independent actions (i.e., the actions in $A_1^V - A_3^V$ or $A_3^V - A_1^V$) in w and w' are interleaved in the same order. But, from the definition of parallel composition it is obvious that these actions can occur in any order in $P_2 \| P_3$ and we can then choose σ' such that $w = w'$. □

It is important to try to find the coarsest equivalence that preserves fair reachability and is substitutive with respect to parallel composition. The following result shows that this coarsest equivalence relation is indeed reach-trace equivalence.

Theorem 14. *Given two IOA P_1 and P_2 with $A_1^I = A_2^I$ and $A_1^O = A_2^O$. If $P_1 =_r P_2$, but $P_1 \neq_{rt} P_2$, then there exists an IOA P_3, compatible with P_1 and P_2 such that $P_1 \| P_3 \neq_r P_2 \| P_3$.*

Proof. Given that P_1 and P_2 are reachability equivalent, but not reach-trace equivalent, there must exist a state $x \in S_1$ and a trace w such that $\langle w, x \rangle \in FairRT(P_1)$ but for all states $y \in S_2$, such that $x =_s y$, we have $\langle w, y \rangle \notin FairRT(P_2)$ or vice versa (switching P_1 and P_2). Without loss of generality we assume that indeed P_1 has such a reach-trace that cannot be simulated by P_2 and let σ_1 be a fair execution of P_1 such that $Tr(\sigma_1) = w$ and $last(\sigma_1) = x$. Let n be the length of w and let $w = \langle a_1, \ldots, a_n \rangle$.

We will now construct an IOA $P_3 = \langle S_3, A_3, R_3^I, z_0 \rangle$ whose only fair trace is w and show that $P_1 \| P_3$ is then indeed not reachability equivalent to $P_2 \| P_3$. We choose

- $S_3 = \{z_i \mid 0 \leq i \leq n+1\}$,
- $A_3^I = A_1^O$, $A_3^O = A_1^I$, and $A_3^H = \{\tau\}$, and
- we choose the transition relation of P_3 as follows

$$\begin{aligned} R_3^I = &\{(z_i, a_{i+1}, z_{i+1}) \mid 0 \leq i < n\} \\ &\cup \{(z_i, b, z_{n+1} \mid 0 \leq i < n, b \in A_3^I \cup A_3^H\} \\ &\cup \{(z_n, b, z_{n+1} \mid b \in A_3^I\} \cup \{(z_{n+1}, \tau, z_{n+1}\}. \end{aligned}$$

It is easy to see that P_3 is compatible with P_1 and P_2, the only stable state of P_3 is z_n, and that the only fair non-divergent execution of P_3 is $\sigma_3 = \langle z_0, a_1, \ldots, a_n, z_n \rangle$, with $Tr(\sigma_3) = w$. We then have that σ_1 is compatible with σ_3 and, by Theorem 12, we then find a finite fair execution σ of $P_1 \| P_3$ such that $\sigma \downarrow P_1 = \sigma_1$ and $\sigma \downarrow P_3 = \sigma_3$. We have that $last(\sigma) = x \| z_n$ is fairly reachable in $P_1 \| P_3$.

We now show by contradiction that there is no fairly reachable state $y \| z$ in $P_2 \| P_3$ such that $x \| z_n =_s y \| z$. Assume then that there does exist such a fairly reachable state $y \| z$. Note, first of all that we have $x =_s y$ and $z_n =_s z$ and then we must have $z = z_n$ since the other states of P_3 are all unstable and thus not fairly reachable. Now, let σ' be a fair execution such that $last(\sigma') = y \| z_n$. Such an execution must exist, since we assume $y \| z_n$ is fairly reachable. By Theorem 11, we have that $\sigma_2 = \sigma' \downarrow P_2$ is a fair execution of P_2 and $\sigma_3' = \sigma' \downarrow P_3$ is a fair execution of P_3, such that σ_2 is compatible with σ_3'. Both these executions must be non-divergent and then σ_3' must be σ_3 since this is the only fair, non-divergent execution of P_3. The compatibility of σ_2 and σ_3 then gives us that $Tr(\sigma_2)$ is w. We then have that σ_2 is a finite fair execution of P_2 with trace w that ends in a state y such that $x =_s y$. But this is a contradiction with the fact that P_1 and P_2 are not reach-trace equivalent. □

We have seen that reach-trace equivalence is the coarsest equivalence relation on IOA that preserves reachability and is substitutive with parallel composition and hiding. This means that the set of reach-traces of an IOA can be used to completely characterise its reachable states and its reachable states in all possible finite compositions. It is then natural to ask what the most compact way of representing this information is. In other words, we ask, given an IOA P, what is the smallest IOA P' such that P' is reach-trace equivalent to P. Unfortunately, since the set of fair traces is a regular language over the set of actions A, we conjecture that finding the minimal IOA with a certain set of reach-traces is equivalent to finding the minimal non-deterministic finite automaton (NFA) that accepts a particular language over A. However, the problem of minimising a NFA is NP-hard [47] and we then conjecture that minimising an IOA with respect to its fair reach-traces is also NP-hard.

4.7.3 Weak bisimulation

As it is often infeasible to find a minimal representation of an IOA with respect to reach-trace equivalence, we now consider an equivalence relation that is finer than reach-trace

equivalence, but which is computable in polynomial time and can be used to efficiently reduce the size of an IOA while preserving its set of reach-traces. The equivalence we use is *weak bisimulation* and is based on *observational equivalence* for LTSs, with an extra clause to deal with stability.

Two states are weakly bisimilar if they can simulate each other's observable behaviour. To characterise this observable behaviour we introduce *weak transitions*.

Definition 39. *Given an IOA P with states S and actions A, there is an* internal transition *from $x \in S$ to $y \in S$, denoted $x \xrightarrow{A^H} y$, if there exists an internal action $a \in A^H$ such that (x, a, y) is in R^I.*

There is a weak internal transition *from x to y, written $x \longrightarrow\!\!\!\!\!\rightarrow y$, if there are states $x_0, \ldots, x_n$ such that $x_i \xrightarrow{A^H} x_{i+1}$ for all $0 \leq i < n$ and $x_0 = x$ and $x_n = y$. The weak internal transition relation $\longrightarrow\!\!\!\!\!\rightarrow$ is the transitive and reflexive closure of $\xrightarrow{A^H}$. Note that for all states $x \in S$ we have $x \longrightarrow\!\!\!\!\!\rightarrow x$.*

There is a weak transition *from x to y labelled $a \in A^I \cup A^O$, written $x \xrightarrow{a}\!\!\!\!\!\rightarrow y$, whenever there are states $x', y' \in S$ such that $x \longrightarrow\!\!\!\!\!\rightarrow x'$, $x' \xrightarrow{a} y'$, and $y' \longrightarrow\!\!\!\!\!\rightarrow y$.*

We now define weak bisimulation.

Definition 40. *Given states S, actions A, and an interactive transition relation $R^I \subset S \times A \times S$, an equivalence relation $\mathcal{E}$ on S is a* weak bisimulation *with respect to S, A, and R^I, if for any pair of states $x \mathcal{E} y$ and any action $a \in A$ we have*

$$\forall x' \in S \cdot \left(x \longrightarrow\!\!\!\!\!\rightarrow x', \neg(x \mathcal{E} x') \implies \exists y' \in S \cdot y \longrightarrow\!\!\!\!\!\rightarrow y', x' \mathcal{E} y'\right), \tag{4.5}$$

$$\forall x' \in S \cdot \left(x \xrightarrow{a}\!\!\!\!\!\rightarrow x' \implies \exists y' \in S \cdot y \xrightarrow{a}\!\!\!\!\!\rightarrow y', x' \mathcal{E} y'\right), \tag{4.6}$$

$$\forall x' \in S \cdot \left(x \longrightarrow\!\!\!\!\!\rightarrow x', st(x') \implies \exists y' \in S \cdot y \longrightarrow\!\!\!\!\!\rightarrow y', st(y')\right), \text{ and} \tag{4.7}$$

$$\forall x, y \in S \cdot (st(x), st(y) \implies x =_s y). \tag{4.8}$$

We say two states x and y are weakly bisimilar in P, denoted $x \approx_P y$, if there exists a weak bisimulation that relates x and y. We leave out the subscript when clear from context. For any IOA, weak bisimilarity itself is the largest weak bisimulation.

Two comparable IOA P_1 and P_2 are weakly bisimilar*, written $P_1 \approx P_2$ if their initial states are weakly bisimilar with respect to the disjoint union of P_1 and P_2.*

An equivalence relation that satisfies conditions (4.5) and (4.6) is an observational equivalence as introduced by Milner [35]. Condition (4.7) on the other hand appears in the definition of weak bisimulation for IMCs [23]. This condition is necessary since we want weak bisimulation to preserve the fair traces of an IOA. Finally, condition (4.8) ensures that weak bisimulation preserves the fairly reachable states up to $=_s$. This condition is only applied to stable states, since only stable states are fairly reachable. To underline the importance of condition (4.7), consider the two IOA in Figure 4.9. It is clear that the relation $\mathcal{E} = \{(x, x), (x, y), (y, x), (y, y)\}$ satisfies the first two conditions, however it does not satisfy condition (4.7) since state x may reach a stable state (itself), while state y cannot. For the fair executions of P_1 and P_2 we find

$$FairEx(P_1) = \{\langle x \rangle, \langle x, \bot \rangle\}$$

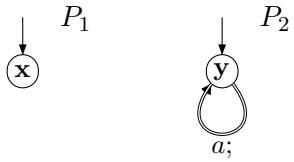


Figure 4.9: Two IOA that are distinguished by the third weak bisimulation condition. The action a is internal. We have $x =_s y$.

and

$$FairEx(P_2) = \{\langle (y,a)^i, y, \bot\rangle \mid i \in \mathbb{N}\}.$$

Note that the execution $\langle y\rangle$ is not fair for P_2 as it does not end in a stable state. For the fair reach-traces we now have

$$FairTr(P_1) = \{\langle \epsilon, x\rangle, \langle \epsilon, \bot\rangle\}$$

and

$$FairTr(P_2) = \{\langle \epsilon, \bot\rangle\}.$$

In other words, P_1 may reach a stable state, while for P_2 we see that time divergence must occur. It then makes sense that these two IOA are not weakly bisimilar.

For the following two lemmas we consider an IOA $P = \langle S, A, R^I, \hat{x}\rangle$. Although not explicitly stated, weak bisimilarity preserves stability. That is, if a state x is stable, then any bisimilar state y can reach a stable state in the same equivalence class with internal transitions and y has no internal transition to other equivalence classes.

Lemma 8. *Given two states $x, y \in S$ such that $x \approx y$,*

$$st(x) \text{ implies } \exists y' \cdot y \longrightarrow\!\!\!\!\!\rightarrow y', st(y') \text{ and } \nexists y'' \cdot y \longrightarrow\!\!\!\!\!\rightarrow y'', y \not\approx y''.$$

Proof. Easy given that x must simulate the internal weak transitions of y. □

We wish to show that weak bisimilarity preserves the fair reach-traces of an IOA. To do this we first show that weak bisimilarity preserves the fair reach-traces of states of an IOA.

Lemma 9. *Given two states $x, y \in S$ such that $x \approx y$ we have that every fair reach-trace of x can be "simulated" by y.*

$$\forall (w, x') \in FairRT(x) \cdot \exists (w, y') \in FairRT(y) \cdot x' = y' = \bot \vee x' \approx y'.$$

Proof. We prove Lemma 9 by induction on the length of the reach-trace, that is the number of visible actions in the trace.

For the empty, divergent reach-trace $(\epsilon, \bot)$ we have that this is a fair reach-trace for any state, so also for y. For an empty non-divergent reach-trace (ϵ, x') with $x' \neq \bot$

we have $x \longrightarrow\!\!\!\!\!\rightarrow x'$ and $st(x')$. For the case that $x \approx x'$, we have $y \approx x'$ and then, by Lemma 8, there is a state y' such that $y \longrightarrow\!\!\!\!\!\rightarrow y'$, $st(y')$, and $x' \approx y'$ which means (ϵ, y') is a fair reach-trace of y. For the case that $x \not\approx x'$, (4.5) dictates that there is a state y' such that $y \longrightarrow\!\!\!\!\!\rightarrow y'$ and $x' \approx y'$. Now, Lemma 8 again gives us that there exists a stable state y'' such that $y' \longrightarrow\!\!\!\!\!\rightarrow y''$ and $y'' \approx x'$. Now we once more have that $\langle \epsilon, y'' \rangle$ is a fair reach-trace of y.

For a non-empty reach-trace $\langle a \circ w, x' \rangle$ where $a \circ w$ is a word of length $n+1$ we use as our induction assumption that, for any states $x'' \approx y''$ and any word w' of length n, if $\langle w', x' \rangle$ is a fair reach-trace of x'' then there exists a fair reach-trace $\langle w', y' \rangle$ of y'' such that $x' \approx y'$ or $x' = y' = \bot$. We now show that under this induction assumption there also exists a corresponding fair reach-trace of $\langle a \circ w, y' \rangle$ of y. Given that $\langle a \circ w, x' \rangle$ is a fair reach-trace of x, there must exist an execution σ starting in x and ending in x' such that the first visible transition of σ is labelled a. We have

$$\sigma = \langle x_1, b_1, x_2, b_2, \ldots, x_m, a \rangle \circ \sigma',$$

where $b_i \in A^H$ and σ' is an execution fragment from x'' to x'. It is clear that if σ is fair, then also σ' is fair and the reach-trace of σ' is $\langle w, x' \rangle$. Furthermore we have $x \stackrel{a}{\longrightarrow\!\!\!\!\!\rightarrow} x''$.

Now, let's see what this means for the state y. First, it must simulate the weak transition $x \stackrel{a}{\longrightarrow\!\!\!\!\!\rightarrow} x''$. We then find a state y'' such that $y \stackrel{a}{\longrightarrow\!\!\!\!\!\rightarrow} y''$ and $x'' \approx y''$. Now, the induction assumption tells us that y'' has a fair reach-trace $\langle w, y' \rangle$ with $x' \approx y'$. From the weak transition $y \stackrel{a}{\longrightarrow\!\!\!\!\!\rightarrow} y''$ and the fair reach-trace $\langle w, y' \rangle$ of y'' it easily follows that y has an execution ρ with reach-trace $\langle a \circ w, y' \rangle$. The final question is whether this reach-trace is fair, or equivalently whether ρ is fair. Since the shorter behaviour $\langle w, y' \rangle$ is fair, we have that y' is either $\bot$ or stable. Since ρ is also finite, the only remaining way in which it can be unfair is if it visits a divergent state but does not end in $\bot$. However, the fact that y' is either stable or $\bot$ immediately makes this impossible (in the former case no state along ρ can be divergent, in the latter case visiting a divergent state is not a problem). □

We can now show that weak bisimilarity indeed preserves the fair reach-traces of an IOA.

Theorem 15. *Given two weakly bisimilar IOA P_1 and P_2 we have that, for any fair reach-trace (w, x) of P_1 there exists a fair reach-trace (w, y) of P_2 such that $x = y = \bot$ or $x =_s y$ with respect to the disjoint union of P_1 and P_2.*

Proof. Theorem 15 follows directly from Lemma 9 applied to the disjoint union of P_1 and P_2. Since x and y are fairly reachable, if they are not equal to $\bot$, they must be stable. Condition 4.8 and the fact that $x \approx y$ then gives us that $x =_s y$. □

Weak bisimulation allows us to find a smaller representation of an IOA by collapsing sets of weakly bisimilar states into a single state.

Definition 41. *For an IOA P with state space S and a state $x \in S$, we write $[x]_\approx$ for the equivalence class with respect to weak bisimulation that contains x, i.e,*

$$[x]_\approx = \{y \in S \mid x \approx y\}.$$

Definition 42. *Given an IOA* $P = (S, A, R^I, \hat{x})$*, its* quotient under weak bisimulation *is an IOA* $[P]_{\approx} = (\bar{S}, \bar{A}, \bar{R}^I, \bar{\hat{x}})$ *with* $\bar{A} = (A^I, A^O, \{\tau\})$ *where states and transitions are defined inductively as follows for* $i \in \mathbb{N}$*:*

$$
\begin{aligned}
\bar{S}_0 =& \{[\hat{x}]_{\approx}\} \\
\bar{R}^I_i =& \{(C, a, C') \mid C \in \bar{S}_i, C' \in S/\!\approx, a \in A^I \cup A^O, \exists x \in C, x' \in C' \cdot x \xrightarrow{a}\!\!\!\!\rightarrow x'\} \cup \\
& \{(C, \tau, C') \mid C \in \bar{S}_i, C' \in S/\!\approx, \exists x \in C, x' \in C' \cdot x \longrightarrow\!\!\!\!\rightarrow x' \wedge C \neq C'\} \cup \\
& \{(C, \tau, C) \mid C \in \bar{S}_i, \forall x \in C \cdot \\
& \qquad (\nexists C'' \in S/\!\approx, x' \in C'', a \in A^O_P \cdot x \xrightarrow{a}\!\!\!\!\rightarrow x') \wedge \\
& \qquad (\nexists C'' \in S/\!\approx, x' \in C'' \cdot x \longrightarrow\!\!\!\!\rightarrow x' \wedge C \neq C'') \wedge \\
& \qquad (\nexists x' \in S \cdot x \longrightarrow\!\!\!\!\rightarrow x' \wedge st(x'))\} \\
\bar{S}_{i+1} =& \{C' \mid C' \in (S/\!\approx) \setminus \textstyle\bigcup_{j=0}^{i} \bar{S}_j, \exists C \in \bar{S}_i \cdot C \xrightarrow{a} C'\}.
\end{aligned}
$$

We then have $\bar{S} = \bigcup_{i=0}^{\infty} S_i$ *and* $\bar{R}^I = \bigcup_{i=0}^{\infty} \bar{R}^I_i$*. For the initial state we find that* $\bar{\hat{x}} = [\hat{x}]_{\approx}$*. Note that we have* $\bar{S} \subset S/\mathcal{E}$*.*

We will give an in-depth explanation of the weak bisimulation quotient for I/O-IMCs in Chapter 5. We will also see that the weak bisimulation quotient for I/O-IMCs can be computed in polynomial time and space, and then the same goes for IOA, since an IOA can be interpreted as an I/O-IMC.

4.8 Confluence and determinism

Whenever two transitions are enabled in an IOA, the choice between these two transitions is non-deterministic, i.e., we do not know which of the transitions will occur. In the context of I/O-IMCs it will be crucial to know whether an IOA displays such non-determinism or not (see Chapter 7).

4.8.1 Confluence

The notion of confluence was introduced by Milner in the context of CCS [35]. In this section we adapt these notions to IOA. An IOA is weakly confluent if, whenever two actions are enabled in a state of the IOA it does not matter in which order these two actions are performed. We will not need to discuss *strong confluence* (see Milner [35]) in this thesis.

Definition 43 ([35])**.** *An IOA* $P = (S, A, R^I, \hat{x})$ *is* weakly confluent *if, for all states* $x_1, x_2, x_3 \in S$*, distinct pairs of visible actions* $a, b \in A^V$*, we have*

$$x_1 \longrightarrow\!\!\!\!\rightarrow x_2, x_1 \longrightarrow\!\!\!\!\rightarrow x_3 \text{ implies } \exists x_4, x_5 \cdot x_2 \longrightarrow\!\!\!\!\rightarrow x_4, x_3 \longrightarrow\!\!\!\!\rightarrow x_5, x_4 \approx x_5, \tag{4.9}$$

$$x_1 \xrightarrow{a}\!\!\!\!\rightarrow x_2, x_1 \longrightarrow\!\!\!\!\rightarrow x_3 \text{ implies } \exists x_4, x_5 \cdot x_2 \longrightarrow\!\!\!\!\rightarrow x_4, x_3 \xrightarrow{a}\!\!\!\!\rightarrow x_5, x_4 \approx x_5, \tag{4.10}$$

$$x_1 \xrightarrow{a}\!\!\!\!\rightarrow x_2, x_1 \xrightarrow{b}\!\!\!\!\rightarrow x_3 \text{ implies } \exists x_4, x_5 \cdot x_2 \xrightarrow{b}\!\!\!\!\rightarrow x_4, x_3 \xrightarrow{a}\!\!\!\!\rightarrow x_5, x_4 \approx x_5, \tag{4.11}$$

$$x_1 \xrightarrow{a}\!\!\!\!\rightarrow x_2, x_1 \xrightarrow{a}\!\!\!\!\rightarrow x_3 \text{ implies } \exists x_4, x_5 \cdot x_2 \longrightarrow\!\!\!\!\rightarrow x_4, x_3 \longrightarrow\!\!\!\!\rightarrow x_5, x_4 \approx x_5, \tag{4.12}$$

where x_4 and x_5 are of course states in S.

Note that the above definition uses single actions a and b, whereas Milner uses action sequences. However, it is easy to see that the two definitions are equivalent. We now note a few important properties of weak confluence, which have been shown by Milner.

Theorem 16 ([35])**.** *For a weakly confluent IOA $P_1 = (S, A, R^I, \hat{x})$ we have that*

1. *for any weakly confluent IOA P_2 that is compatible with P_1, $P_1 \| P_2$ is weakly confluent*[3]*,*
2. *for any subset of output actions $B \subset A^O$, $P_1 \backslash B$ is weakly confluent,*
3. *for any pair of states $x, y \in S$, $x \longrightarrow\!\!\!\!\!\rightarrow y$ implies $x \approx y$, and*
4. *for any IOA P_2 we have $P_1 \approx P_2$ implies that P_2 is also weakly confluent.*

It will be useful to consider the confluence properties for different pairs of actions in isolation.

Definition 44. *Given an IOA $P = (S, A, R^I, \hat{x})$ and a distinct pair of visible actions $a, b \in A^V$, we say that P is* weakly confluent with respect to a and b *if, for all states $x_1, x_2, x_3 \in S$, we have that the property* (4.11) *holds.*

4.8.2 Determinism

In the context of I/O-IMCs it will be extremely important to know whether a model is *deterministic*[4] or not. We say that a closed IOA is *weakly* deterministic if it has only one non-divergent reach-trace, but since this is difficult to verify we use the following definition which is similar to the definition of weak confluence.

Definition 45. *A closed IOA $P = (S, A, R^I, \hat{x})$ is* weakly deterministic *if for any states $x_1, x_2, x_3 \in S$, and any pair of output actions $a, b \in A^O$ we have*

$$x_1 \longrightarrow\!\!\!\!\!\rightarrow x_2 \text{ implies } x_1 \approx x_2, \text{ and} \tag{4.13}$$

$$x_1 \xrightarrow{a}\!\!\!\!\!\rightarrow x_2, x_1 \xrightarrow{b}\!\!\!\!\!\rightarrow x_3 \text{ implies } a = b, x_2 \approx x_3. \tag{4.14}$$

For a fixed pair of output actions $a, b \in A^O$, we say the closed IOA P is weakly deterministic with respect to a and b *if* (4.14) *holds.*

Proposition 9. *A closed, weakly deterministic IOA P with no divergent states has exactly one non-divergent fair reach-trace.*

[3] The parallel composition operator for IOA is less general than the one introduce by Milner for LTSs, which is why we need not introduce the notion of *restricted composition* (see [35, pp. 244]). In a sense IOA parallel composition is *restricted* by definition.

[4] Note that the term "determinism" has been used in different ways in different contexts. It is not to be confused with *determinacy* as used by Milner [35]

Proof. This follows from the fact that weak bisimulation preserves fair reach-traces and the fact that weak determinism does not allow choices between different actions or choices between states that are not weakly bisimilar. □

Note that weakly deterministic IOA may have multiple non-divergent fair reach-traces. This is caused by the inherently non-deterministic nature of time divergence. In the following, we will assume that the closed, weakly deterministic IOA are never composed in parallel with divergent IOA. Under this assumption it makes sense to only consider the non-divergent fair reach-traces of such an IOA. As we might expect, the weak bisimulation quotient of a weakly deterministic IOA has a very simple form.

Proposition 10. *Given a closed, weakly deterministic IOA $P = (S, A, R^I, \hat{x})$, we find for its quotient $[P]_{\approx}$ that for any three equivalence classes $D_1, D_2, D_3 \in S/\approx$ and any two output actions $a, b \in A^O$ we have*

$$D_1 \longrightarrow\!\!\!\!\!\rightarrow D_2 \text{ implies } D_1 = D_2, \text{ and} \tag{4.15}$$

$$D_1 \xrightarrow{a} D_2, D_1 \xrightarrow{b} D_3 \text{ implies } a = b, D_2 = D_3. \tag{4.16}$$

Proof. Simple. □

The weak bisimulation quotient of a weakly deterministic IOA P is then simply a single chain of states and transitions which, if the quotient is finite, ends either in an absorbing state with no outgoing transitions or a divergent state.

The connection between weak confluence and weak determinism is as follows. Hiding a set of pairwise weakly confluent actions "preserves" weak determinism.

Proposition 11. *Given a closed IOA $P = (S, A, R^I, \hat{x})$ and a set of output actions $B \subset A^O$, we have that the IOA $P\backslash B$ is weakly deterministic if*

1. *for any two states $x_1, x_2 \in S$ we have that $x_1 \longrightarrow\!\!\!\!\!\rightarrow x_2$ implies $x_1 \approx x_2$,*
2. *P is* weakly confluent *with respect to all pairs of actions $a, b \in B$, and*
3. *P is* weakly deterministic *with respect to all remaining pairs of actions $a, b \in A^O \setminus B$.*

Proof. Simple. □

For a *complete* IOA, we need only to check the first two conditions of the above proposition. We can then make use of the fact that weak confluence is compositional.

Proposition 12. *Given $n \in \mathbb{N}$ pairwise compatible, weakly confluent, IOA $P_1, \ldots, P_n$ and a set of actions B we have that if $(P_1 \| \ldots \| P_n)\backslash B$ is complete, then it is weakly deterministic.*

Proof. Easy. □

Proposition 12 provides a sound way of finding out if a system of IOA is deterministic in polynomial time. However, this method is not complete. That is, we can easily construct IOA $P_1, \ldots, P_n$ which are not all weakly confluent, but whose parallel composition is weakly deterministic. In Chapter 8 we will try, in the setting of I/O-IMCs, to improve on Proposition 12, by developing another sound method of finding out if a system of I/O-IMCs is deterministic with the same time complexity, which is more complete, i.e., has less false negatives. We will see in Chapter 9 why it is so important to find out whether an I/O-IMC is deterministic or not.

4.9 Discussion

This section reviews the material developed in this chapter, and places it in the context of the original and mainstream work on IOA.

4.9.1 Particularities

We have presented the basics of input/output automata, together with meaningful composition operators, equivalence notions, and a notion of confluence that will become particularly relevant in Chapter 8.

Since this chapter has taken strong inspiration from the original IOA work by Lynch and Tuttle [33], the results achieved are not very surprising. Some deviations were needed in order to prepare for the subsequent combination with the Markov chain theory developed in Chapter 3.

First of all we are imposing stronger fairness assumptions. The fairness assumptions of Lynch and Tuttle focus on a scenario where an IOA represents a reactive process with several *tasks* where one task should not indefinitely block progress of the others. We deviate from that treatment on the one hand to accommodate the use of weak bisimulation as an equivalence relation, and on the other hand to deal with time divergence, a phenomenon relating to the occurrence of infinitely many events in a finite amount of time. Concretely, we extend fairness to individual transitions (rather than to sets of actions, i.e., tasks) and secondly we consider not only the trace of actions that occur along a path in the IOA, but also the final state of that path. Finally, we represent infinite traces using the distinguished state $\perp$. A useful consequence of our adaptations is that the fair executions of a closed IOA (closed meaning that the IOA cannot be influenced by its environment) are all finite. Either they reach a state in which no further interactions are possible, or they reach the distinguished state $\perp$ denoting time divergence.

Secondly, we interpret the visible behaviour of an IOA not simply as a series of visible actions (the *trace*) describing the types of events that occur during a fair execution, but in addition we record the final state of the fair execution. This final state is of critical importance when we combine interactive behaviour and Markovian behaviour. This led us to the notion of *reach-trace* equivalence. As computing reach-trace equivalences is impractical, we have also introduced the finer equivalence of weak bisimulation, which we will lift to I/O-IMCs in Chapter 5.

The importance of reach-traces stems from the fact that, in Chapter 6, we will use IOA together with CTMCs in a combined model, where sequences of interactions (modelled as IOA) alternate with Markovian phases (modelled as CTMCs), and where the final state of an interactive phase will be the starting point for the subsequent Markovian phase.

4.9.2 Comparison to process calculi

In IOA, interaction between components is modelled through *asymmetric synchronisation*, in the sense that every action (i.e., every type of event) is controlled by exactly one IOA, which controls when such events occur. The action is an output or internal action for this IOA. For each action there may be zero or more passive component that *react* to events associated with that action. The action is an *input* action for these IOA. This asymmetry is enforced by the compatibility requirements for composing IOA (see Definition 30). In contrast, for process calculi such as Milner's CCS and Hoare's CSP more than one component may control an action and events associated with that action can only occur if all components enable that action [35, 27]. In these process calculi there is no distinction between input and output actions. One very important difference between IOA and CCS/CSP is that for the latter trace equivalence is not a congruence. For more details on the difference between IOA and process calculi we refer to Vaandrager [49].

Another interesting related model is the input/output labelled transition system (IOLTS) formalism as introduced by Tretmans [48]. IOLTSs are very similar to IOA, except that the requirement of input-enabledness is more relaxed than for IOA. To be precise, only *weak* input-enabledness is required. That is, for every state x and input action a of an IOLTS, it is required that there exists a state y such that $x \stackrel{a}{\longrightarrow\!\!\!\!\!\rightarrow} y$ (recall that the input-enabledness condition for IOA requires $x \stackrel{a}{\longrightarrow} y$). This means x may not have an outgoing a-transition, but x can reach, using internal transitions a state x' that has an outgoing a-transition. It is not clear what the implications of this relaxation are on the results of this chapter. We do note that IOLTSs are mainly used to facilitate *model-based testing* in which internal transitions are used to model the freedom of design in a specification [48]. In contrast, when we use I/O-IMCs (see Chapter 5) to model dependable systems in Chapter 9, we use internal transitions only to abstract away from internal behaviour in parallel compositions of I/O-IMCs. In fact, none of the elementary I/O-IMCs used in Chapter 9 have any internal transitions, so the relaxation of the input-enabledness condition is meaningless for these models. However, there may be other applications of I/O-IMCs where it will be useful to considering relaxing the input-enabledness condition as for IOLTSs.

4.9.3 IOA as a graph-based model

In this chapter we have seen that IOA can be represented as a graph with actions on its edges. We have chosen as semantics for IOA the set of fair *reach-traces* of the IOA. IOA come equipped with a natural notion of composition. In terms of the

graph-representation of IOA, composition is achieved by *synchronising* transitions with identical actions and *interleaving* transitions with different actions. The semantics of an IOA composition can be obtained by combining *compatible* reach-traces, where two reach-traces are compatible if they agree on the order of their shared actions. Crucially, this semantics is modular (see Theorems 11 and 12), in the sense that the relationship between syntax and semantics is preserved when composing IOA. Next to CTMCs (see Chapter 3), IOA form the second ingredient of I/O-IMCs both in terms of their (graph-based) syntax (see Chapter 5) and in the context of their semantics (see Chapter 6). Crucially, we will use the modularity results for IOA, to show that the semantics of I/O-IMCs is modular as well.

5

Input/Output Interactive Markov Chains

In this chapter we take up the foundations laid in the preceding two chapters and introduce input/output interactive Markov chains (I/O-IMCs). Intuitively, an I/O-IMC describes a system which may change its state either due to interaction with its environment–in the same way as IOA–or stochastically after a certain delay–in the same way as Markov chains. We will focus our attention mainly on the composition operators on I/O-IMCs, together with their impact on properties we are interested in. We will introduce means to construct composite I/O-IMCs by letting smaller I/O-IMCs run in parallel, just as we have done for IOA. We combine the notions of bisimulation for Markov chains and weak bisimulation for IOA to form a natural equivalence relation on I/O-IMCs, and lift several concepts from the IOA setting to I/O-IMCs.

Contribution. The chapter develops the syntactic foundations of I/O-IMCs. Most of this chapter is rooted in previous joint work [4, 5]. However, we here explore the connection between I/O-IMCs and IOA by decomposing the former into its IOA constituents, This allows us to lift many of the concepts introduced for IOA, such as fair reach-traces, confluence, and determinism, directly to I/O-IMC. Finally, we also introduce the notion of *stochastic reachability* which will play a crucial role in the remainder of this thesis as the primary notion of reachability for I/O-IMCs.

5.1 I/O-IMC ingredients

This section presents the basic definition of an I/O-IMC and discusses its building blocks. Formally, an I/O-IMC is described as follows.

Definition 46. *An I/O-IMC P is a five-tuple $\langle S, A, R^I, R^M, \alpha \rangle$, where*

- *The* state space *S is a countable set of states,*

- *The set of* actions A *is a finite set, disjoint from* S*, which is partitioned into the set of* input actions A^I*,* output actions A^O*, and* internal (hidden) actions A^H*,*
- *The* interactive transition relation R^I*, which is a subset of* $S \times A \times S$*,*
- *The* Markovian transition relation R^M*, which is a subset of* $S \times \mathbb{R}_{\geq 0} \times S$*, and*
- *The initial distribution* α *which is a distribution over* S*.*

As for IOA, we require that the I/O-IMC is input-enabled*; every state must have, for every input action, at least one outgoing transition labelled with that input action. That is,*

$$\forall x \in S, a \in A^I \cdot \exists y \in S \cdot (x, a, y) \in R^I. \tag{5.1}$$

We will use x, y, z as well as the indexed versions $x_i, y_i, z_i, i \in \mathbb{N}$ to indicate states in S. We will use a, b, c as well as their indexed versions to indicate actions. We will use $\lambda, \mu, \kappa, \nu$ as well as their indexed versions to indicate *rates*, taken from $\mathbb{R}_{\geq 0}$, in Markovian transitions. Whenever it is clear from context which I/O-IMC is meant, we will use the predicate $x \xrightarrow{a} y$ to denote the existence of a transition (x, a, y) in R^I and the predicate $x \xhookrightarrow{\lambda} y$ to denote the existence of a transition (x, λ, y) in R^M.

We assume that the Markovian transition relation contains no parallel edges. This means that for all $x, y \in S$ we require that there is at most one transition $(x, \cdot, y)$ in R^M. Note that any I/O-IMC with parallel Markovian edges can easily be represented without parallel Markovian edges by replacing the parallel edges with a single Markovian transition whose rate is the sum of the rates of the parallel transitions it replaces.

Example 17. *As an example, Figure 5.1 shows an I/O-IMC model of a generic repairable component. It might be a processor in a computer system, a pump in a reactor cooling system, or a tire on a car. We only model the* failure behavior *of the component, i.e., how the component may break down and how it may subsequently be repaired. Ellipses denote the possible states of the components. Single arrows denote Markovian transitions, double arrows interactive transitions. The actions of interactive transitions are embellished with a question-mark when the action is an input action, an exclamation mark when it is an output action and a semi-colon if the action is an internal action. The small box labeled one shows that the initial distribution assigns probability one to state "up". Unless explicitly noted we will assume that each action in an I/O-IMC appears at least on one transition. Under this assumption the Figure 5.1 completely specifies the I/O-IMC. To be precise, we have*

$$\begin{aligned}
S &= \{\mathbf{up}, \mathbf{failing}, \mathbf{down}, \mathbf{recovering}\},\\
A^I &= \{repair\}, A^O = \{fail, recover\}, A^H = \emptyset\\
R^I &= \{(\mathbf{up}, repair, \mathbf{up}), (\mathbf{failing}, repair, \mathbf{up}), (\mathbf{failing}, fail, \mathbf{down}),\\
&\quad (\mathbf{down}, repair, \mathbf{recovering}), (\mathbf{recovering}, repair, \mathbf{recovering}),\\
&\quad (\mathbf{recovering}, recover, \mathbf{up})\}\\
R^M &= \{(\mathbf{up}, \lambda, \mathbf{failing})\}\\
\alpha &= \{(\mathbf{up}, 1), (\mathbf{failing}, 0), (\mathbf{down}, 0), (\mathbf{recovering}, \mathbf{0})\}.
\end{aligned}$$

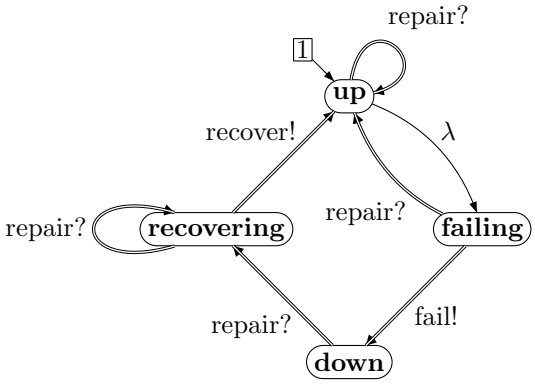


Figure 5.1: Example of an I/O-IMC.

In the remainder of this section we will consider an I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$ with input actions A^I, output actions A^O, and internal actions A^H.

5.1.1 State space

The *state space* of an I/O-IMC is discrete and represents the different states of a component. For instance, the I/O-IMC from Example 17 has states *operational*, *failing*, *down*, and *recovering*. The state space of an I/O-IMC may also be infinite. Consider, for instance, an I/O-IMC that models an unbounded queue. The states of this I/O-IMC would then count the number of objects in the queue. We would then have $S = \mathbb{N}$. We do not consider uncountably large state spaces.

In contrast to IMCs (see [24]) and other process algebras, we do not use states to denote the dynamics of the system as well as its state. The state space S of an I/O-IMC is a subset of our set of all states S_{all} (cf. Section 2.1), which comes equipped with a notion of composition which we will use in Section 5.3 when we discuss parallel composition for I/O-IMCs. In general, the state spaces of different I/O-IMCs may overlap.

5.1.2 Actions

Similar to IOA, I/O-IMCs are event-based models. The actions of an I/O-IMC give names to these events. For the I/O-IMC in Example 17 we have actions *fail*, *repair*, and *recover*. It is important to note that the events in an I/O-IMC occur *instantaneously*. Also, during the lifetime of an I/O-IMC multiple events with the same name may happen. Since we have input, output, and internal actions, we also have input, output and internal events. Input actions describe events that are not controlled by the I/O-IMC itself, but do influence it. Output actions on the other hand are controlled by the I/O-IMC itself and they may influence other I/O-IMCs. We will see how different I/O-IMCs can influence each other via input and output actions when we discuss parallel composition in Section 5.3. Finally, internal or hidden actions are controlled by the I/O-IMC itself,

but they cannot influence other I/O-IMCs or be observed. As for IOA, we say that input or output actions are *visible* and output or internal actions are *locally-controlled*.

Definition 47. *An action $a \in A$ is called* visible *for P if it is either an input action or an output action of P. We then define the set of visible actions by*

$$A_P^V = A^I \cup A^O.$$

An action $a \in A$ is called locally-controlled *by P if its either an output or internal action of P. We then define the set of locally-controlled actions by*

$$A_P^C = A^O \cup A^H.$$

We will leave out the subscript whenever it is clear from context which I/O-IMC is meant.

5.1.3 Interactive transition relation

We will use the interactive transition relation to describe the possible interactions between components. For a state $x \in S$ the interactions that may occur when the I/O-IMC occupies x are described by the IOA *rooted at* x. This is simply the IOA obtained by ignoring the Markovian transitions in the I/O-IMC and choosing the initial state to be x.

Definition 48. *Given an I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$ and a state $x \in S$, the* IOA rooted at x *is given by $IOA_P(x) = \langle S, A, R^I, x \rangle$. For the sake of simplicity, we sometimes omit states and transitions of $IOA_P(x)$ that are unreachable. Whenever it is clear from context which I/O-IMC is meant, the subscript is omitted.*

The possible interactions, that may happen when the I/O-IMC occupies state $x \in S$, are now exactly the fair reach-traces of $IOA(x)$ as defined in Section 4.4. That is, given a state $x \in S$ we find a set of pairs (w, y) such that the I/O-IMC can *reach* y through a sequence of interactions labelled with the actions w. We also find pairs $(w, \perp)$ that denote that, after a sequence of interactions labelled with the actions w, the I/O-IMC may experience time divergence.

Example 18. *Consider the state **failing** of the repairable component I/O-IMC from Example 17. By looking at the IOA rooted at the state **failing** we see that the following fair reach-traces are possible in this state (among others):*

$$(\langle \mathit{repair} \rangle, \mathbf{up})$$
$$(\langle \mathit{repair}, \mathit{repair} \rangle, \mathbf{up})$$
$$(\langle \mathit{fail} \rangle, \mathbf{down})$$
$$(\langle \mathit{fail}, \mathit{repair}, \mathit{recover} \rangle, \mathbf{up}).$$

Of course, divergent reach-traces, such as

$$(\langle \mathit{fail}, \mathit{repair} \rangle, \perp)$$

are also possible.

5.1.4 Markovian transition relation

The Markovian transition relation of an I/O-IMC describes changes in state that may occur spontaneously after the I/O-IMC occupies a state for some time. The label of a Markovian transition is called the transition *rate*. As for CTMCs we can describe the infinitesimal transition probabilities of an I/O-IMC using a matrix.

Definition 49. *The* Q-matrix *of I/O-IMC P is the matrix $Q : \mathbb{R}_{\geq 0}^{|S|\times|S|}$ with entries $\{q_{x,y} \mid x, y \in S\}$ with*

$$q_{x,y} = \begin{cases} \lambda, & \text{if } x \neq y, x \xhookrightarrow{\lambda} y, \\ 0, & \text{if } x \neq y, \nexists \lambda \cdot x \xhookrightarrow{\lambda} y, \\ -\sum_{y \in S, y \neq x} q_{x,y}, & \text{if } x = y. \end{cases} \tag{5.2}$$

Recall that for a pair of states x, y there is at most one Markovian transition from x to y. For convenience we will use the notation $q_x = -q_{x,x}$.

Intuitively, we have that a Markovian transition $x \xhookrightarrow{\lambda} y$ from state x to state y with rate λ, means that $q_{x,y} = \lambda$. Just like for Markov chains we will see (in Chapter 6) that the probability that the I/O-IMC will "jump" from state x to state y in a time-interval of length h is $\lambda h + o(h)$. As for Markov chains, the infinitesimal generator matrix of an I/O-IMC is used to compute its finite jump probabilities. However, unlike a Markov chain, the finite jump probabilities are not completely determined by Q. We will go into more detail in Chapter 6.

For the I/O-IMC from Example 17 the intuition is that it will move from state **up** to state **failing** after a delay that is exponentially distributed with rate λ, given that no *repair* events occur[1]. In Chapter 6 we will give a complete semantics to I/O-IMCs from which we can derive these delay distributions.

5.1.5 Initial distribution

The initial distribution of an I/O-IMC dictates which state the I/O-IMC occupies at time-point zero. For our running example we have that the I/O-IMC starts in state **up** with probability one.

5.2 Classification of states

We are now in the position to lift the notions of stability and divergence (as introduced for IOA in Section 4.2) to I/O-IMCs.

Definition 50. *A state $x \in S$ of P is*

- stable*, if it has no outgoing interactive transitions, and*

[1]Due to the memoryless nature of the exponential distribution and the fact that any *repair* event takes the I/O-IMC back to state **up**, the statement actually still holds if finitely many *repair* events happen.

- divergent, *if there exists no interactive path from x to any stable state.*

Note that a state x is stable respectively divergent if and only if it is stable respectively divergent in $IOA(x)$.

For our running example we have that states **up** and **down** are stable, states **failing** and **recovering** are unstable and no states are divergent.

5.3 Parallel composition

We now consider the possibility of letting two or more I/O-IMCs "execute in parallel". Intuitively, the Markovian behaviour of these two I/O-IMCs is independent and Markovian transitions are thus interleaved, while the interactive transitions are synchronized via the shared alphabet of the two I/O-IMCs. As for IOA, we only consider the parallel composition of *compatible* I/O-IMCs.

Definition 51. *Given the following two I/O-IMCs $P_1 = \langle S_1, A_1, R_1^I, R_1^M, \alpha_1\rangle$ and $P_2 = \langle S_2, A_2, R_2^I, R_2^M, \alpha_2\rangle$, we say P_1 and P_2 are* compatible *if*

1. *they do not share any output actions,*

$$A_1^O \cap A_2^O = \emptyset,$$

and

2. *their internal actions are unique,*

$$A_1^H \cap (A_2^I \cup A_2^O \cup A_2^H) = \emptyset$$

and

$$A_2^H \cap (A_1^I \cup A_1^O \cup A_1^H) = \emptyset.$$

A set of I/O-IMCs is compatible if the I/O-IMCs in the set are pairwise compatible.

The parallel composition is found by synchronizing transitions labelled by actions from the shared alphabet and interleaving all other transitions (including Markovian transitions).

Definition 52. *The* parallel composition *of two compatible I/O-IMCs P_1 and P_2 is the I/O-IMC $P_1 \| P_2 = \langle S_1 \| S_2, A, R^I, R^M, \alpha\rangle$, where*

- *the state space is*

$$S_1 \| S_2 = \{x \| y \mid x \in S_1, y \in S_2\},$$

- *the actions are given by*

$$A^O = A_1^O \cup A_2^O, \quad A^I = A_1^I \cup A_2^I \setminus A^O, \quad A^H = A_1^H \cup A_2^H,$$

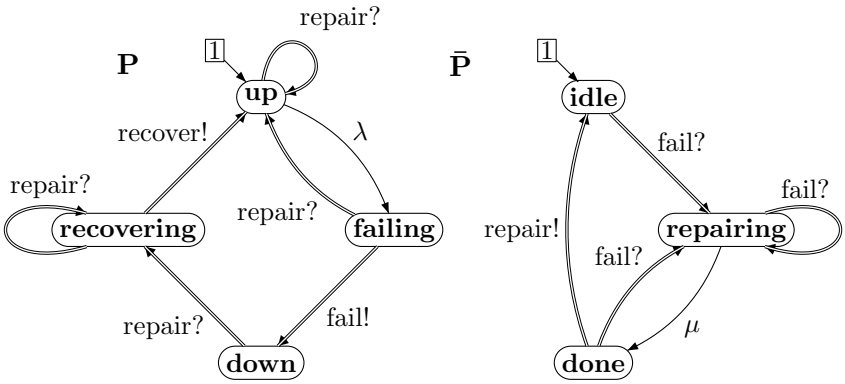


Figure 5.2: Example of two compatible I/O-IMCs.

- *the interactive transition relation is found by synchronising on shared actions*

$$\begin{aligned} R^I = &\{(x_1\|x_2, a, y_1\|y_2) \mid a \in A_1 \cap A_2, (x_1, a, y_1) \in R_1^I, (x_2, a, y_2) \in R_2^I\} \\ &\cup \{(x_1\|x_2, a, y_1\|x_2) \mid a \in A_1 - A_2, (x_1, a, y_1) \in R_1^I\} \\ &\cup \{(x_1\|x_2, a, x_1\|y_2) \mid a \in A_2 - A_1, (x_2, a, y_2) \in R_2^I\}, \text{ and} \end{aligned}$$

- *the Markovian transition relation is found by interleaving*

$$\begin{aligned} R^M = &\{(x_1\|x_2, \lambda, y_1\|x_2) \mid (x_1, \lambda, y_1) \in R_1^M\} \\ &\cup \{(x_1\|x_2, \lambda, x_1\|y_2) \mid (x_2, \lambda, y_2) \in R_2^M\}, \text{ and} \end{aligned}$$

- *the initial distribution is given by, for all* $x\|y \in S$,

$$\alpha(x\|y) = \alpha_1(x)\alpha_2(y).$$

Example 19. *As an example we consider again the repairable component from Example 17, which we will compose in parallel with an I/O-IMC model of a repairman. The two I/O-IMCs are shown in Figure 5.2. The repairman is idle until the component fails. It then starts repairing and, after an exponentially distributed delay with rate* μ*, the component is repaired. It is easy to confirm that the I/O-IMCs* P *and* $\bar{P}$*, depicting the repairable component and the repairman respectively, are indeed compatible. Their parallel composition* $\tilde{P}$ *is shown in Figure 5.3*

In the remainder of this section we will consider two compatible I/O-IMCs $P = \langle S, A, R^I, R^M, \alpha\rangle$ and $\bar{P} = \langle \bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha}\rangle$ and their parallel composition $\tilde{P}$.

Parallel composition for I/O-IMCs is closely related to parallel composition for IOA and Markov chains. As a first result we find that the IOA rooted at a state $x\|\bar{x}$ of $\tilde{P}$ is exactly the parallel composition of the IOA rooted at x and $\bar{x}$.

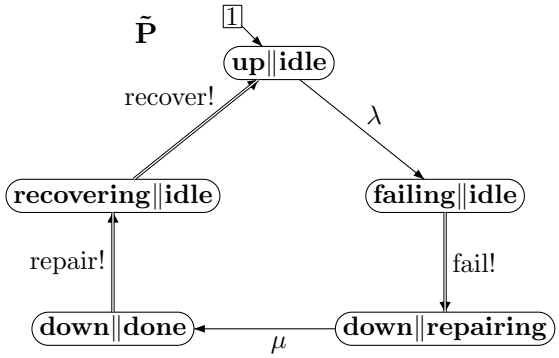


Figure 5.3: Example of a composed I/O-IMC.

Proposition 13. *Given two states $x, \bar{x}$ in S respectively $\bar{S}$ we have that the IOA rooted at x and $\bar{x}$ are also compatible. Moreover the IOA rooted at $x\|\bar{x}$ is the parallel composition of the IOA rooted at x respectively $\bar{x}$, i.e.,*

$$IOA_{\tilde{P}}(x\|\bar{x}) = IOA_P(x)\|IOA_{\bar{P}}(\bar{x}).$$

Proof. The proposition follows directly from the respective definitions. □

An immediate result of Proposition 13 is that I/O-IMCs share the modularity of stability and divergence with IOA.

Corollary 10. *Given two states $x, \bar{x}$ in S respectively $\bar{S}$ we have that*

- *$x\|\bar{x}$ is stable in $\tilde{P}$ if and only if both x and $\bar{x}$ are stable in P respectively $\bar{P}$ and*
- *$x\|\bar{x}$ is non-divergent if both x and $\bar{x}$ are non-divergent. Note that the reverse may not hold.*

Similarly, the infinitesimal generator of $\tilde{P}$ can be found by taking the cross-product of the infinitesimal generator functions of P and $\bar{P}$.

Proposition 14. *For states $x, y \in S_\perp$ and states $\bar{x}, \bar{y} \in \bar{S}_\perp$ we have*

$$\tilde{q}_{x\|\bar{x},y\|\bar{y}} = \begin{cases} q_{x,y} + \bar{q}_{\bar{x},\bar{y}}, & \text{if } x = y, \bar{x} = \bar{y}, \\ q_{x,y}, & \text{if } x \neq y, \bar{x} = \bar{y}, \\ \bar{q}_{\bar{x},\bar{y}}, & \text{if } x = y, \bar{x} \neq \bar{y}, \\ 0, & \text{if } x \neq y, \bar{x} \neq \bar{y}. \end{cases} \tag{5.3}$$

Proof. The proposition follows directly from the definitions of parallel composition and Q-matrix of I/O-IMCs. □

5.4 Hiding

As for IOA, we can abstract from actions of an I/O-IMCs by *hiding* them or in other words, by making them internal. When an action is hidden, it can no longer be observed by other I/O-IMCs. As for IOA, we only allow output actions to be hidden.

Definition 53. *Given an I/O-IMC* $P = \langle S, A, R^I, R^M, \alpha \rangle$ *and a set of output actions* $B \subseteq A^O$, hiding *the actions* B *in* P *results in the I/O-IMC* $P \backslash B = \langle S, \bar{A}, R^I, R^M, \alpha \rangle$, *where*

$$\bar{A}^I = A^I, \bar{A}^O = A^O \setminus B, \text{ and } \bar{A}^H = A^H \cup B.$$

Similar to IOA (see Section 4.6), hiding actions does not affect whether states are stable or divergent and does not affect the interactive reachability properties of the I/O-IMC.

5.5 Equivalences

In this section we will discuss several equivalences for I/O-IMCs. We recall that the state equivalence relation $=_s$ tells us which states are indistinguishable (see Section 2.1). Our goal will be to find an equivalence which

1. is a congruence with respect to parallel composition and hiding,
2. preserves the finite jump probabilities (see Chapter 3) of I/O-IMCs with respect to the state equivalence relation $=_s$, and
3. can be computed efficiently.

We will discuss the first point in detail in this section. For the second point we will show several important results, namely that reach-traces and cumulative transition rates are preserved by our equivalences. However, we will postpone the question, whether our equivalences preserve the finite jump probabilities of I/O-IMCs to Chapter 6.

Throughout this section we have left out proofs which are similar to those for IMCs [23] or have been presented in previous work [4, 6, 7].

5.5.1 Isomorphism

Isomorphism is a very strong equivalence which equates two I/O-IMCs only if we can find a one-to-one correspondence between their states that preserves all transitions, the initial distribution and the equivalence relation $=_s$.

Definition 54. *Given the following two I/O-IMCs* $P = (S, A, R^I, R^M, \alpha)$ *and* $\bar{P} = (\bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ *with disjoint state spaces, and identical input and output actions* ($A^I = \bar{A}^I$ *and* $A^O = \bar{A}^O$), *a bijection* f *from* S *to* $\bar{S}$ *is an* isomorphism *if and only if for all states* $x \in S$ *we have:*

- $\alpha(x) = \bar{\alpha}(f(x))$,

- $x \xrightarrow{a} x'$ *if and only if* $f(x) \xrightarrow{a} f(x')$,
- $x \xhookrightarrow{\lambda} x'$ *if and only if* $f(x) \xhookrightarrow{\lambda} f(x')$, *and*
- $st(x)$ *implies* $x =_s f(x)$.

We say that I/O-IMCs P and $\bar{P}$ are isomorphic, written $P \equiv \bar{P}$, if there is an isomorphism from the states of P to the states of $\bar{P}$.

The first three properties of isomorphism for I/O-IMCs are as can be expected. For the fourth property we see that only *stable* isomorphic states need to be equivalent with respect to $=_s$. This technicality will be necessary later on. The intuition is that the identity (w.r.t. $=_s$) of unstable states does not matter, since the probability to occupy an unstable state is in any case zero for any time-point before explosion (see Proposition 19). Isomorphism is interesting theoretically as it is the strongest useful equivalence on graph models. The practical usefulness of isomorphism is doubtful, as determining whether two graphs are isomorphic is in NP. Furthermore, isomorphism is clearly not a congruence with respect to parallel composition and hiding.

5.5.2 Strong Bisimulation

The other two equivalence relations we consider are *bisimulations*. Both of these bisimulations will first be introduced as *state-wise* equivalences defined on the states of a single I/O-IMC. We will later see how these state-wise equivalences can be lifted to equate I/O-IMCs themselves.

We first consider strong bisimulation. We will see that this equivalence is weaker than isomorphism. Strong bisimulation considers cumulated Markovian rates and it uses the maximal progress assumption to abstract away from Markovian transitions that are taken with probability zero and the identity of states (according to $=_s$) that are occupied with probability zero at any time-point (i.e., unstable states as per Proposition 19).

Before defining strong bisimulation, we need to introduce one more notation. Given an I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, a state $x \in S$, and a set of states $C \subseteq S$ we denote the *aggregate* Markovian transition from x into C as $x \xhookrightarrow{\lambda} C$ where

$$\lambda = \sum \{\mu \mid y \in \mathcal{C}, x \xhookrightarrow{\mu} y\}.$$

Definition 55 (Strong bisimulation for states)**.** *Given I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, an equivalence relation $\mathcal{E}$ on S is a* strong bisimulation *on P if and only if for all pairs of states $x_1, x_2 \in S$ such that $x_1 \mathcal{E} x_2$ we have:*

1. *All observable interactive transitions emerging from x_1 can be simulated by x_2:*

$$\forall a \in A^O \cup A^I \cdot x_1 \xrightarrow{a} x_1' \implies x_2 \xrightarrow{a} x_2' \wedge x_1' \mathcal{E} x_2'.$$

2. *All internal interactive transitions emerging from x_1 can be simulated by x_2, possibly with a different internal action:*

$$\forall a \in A^H \cdot x_1 \xrightarrow{a} x_1' \implies \exists b \in A^H \cdot x_2 \xrightarrow{b} x_2' \wedge x_1' \mathcal{E} x_2'.$$

3. *If x_1 is stable, then all Markovian transitions emerging from x_1 can be simulated by x_2:*
$$st(x_1) \implies (\forall C \in S/\mathcal{E} \cdot x_1 \xhookrightarrow{\lambda} C \implies x_2 \xhookrightarrow{\lambda} C)$$

4. *If x_1 is stable, then x_2 must be equivalent to x_1 according to $=_s$*
$$st(x_1) \implies x_1 =_s x_2.$$

We say that two states x_1 and x_2 of P are strongly bisimilar, written $x_1 \sim_P x_2$, if there exists a strong bisimulation $\mathcal{E}$ on P such that $x_1 \mathcal{E} x_2$. We leave out the subscript when clear from context.

Is is straightforward to show that for two strongly bisimilar states x_1 and x_2 we have that x_1 is stable if and only if x_2 is stable.

Theorem 17. *Given an I/O-IMC P, strong bisimilarity on P is the largest strong bisimulation on P.*

Proof. Standard. □

Definition 56. *Given an I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, its* quotient under strong bisimulation *is an I/O-IMC $[P]_\sim = (\bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ with $\bar{A} = (A^I, A^O, \{\tau\})$ where states and transitions are defined inductively as follows for $i \in \mathbb{N}$:*

$$\begin{aligned}
\bar{S}_0 =& \{C \mid C \in S/\sim, \alpha(C) > 0\} \\
\bar{R}_i^I =& \{(C, a, C') \mid C \in \bar{S}_i, C' \in S/\sim, a \in A^V, \exists x \in C, x' \in C' \cdot x \xrightarrow{a} x'\} \cup \\
& \{(C, \tau, C') \mid C \in \bar{S}_i, C' \in S/\sim, b \in A^H, \exists x \in C, x' \in C' \cdot x \xrightarrow{b} x'\} \\
\bar{R}_i^M =& \{(C, \lambda, C') \mid C \in \bar{S}_i, C' \in S/\sim, \exists x \in C \cdot st(x) \wedge x \xhookrightarrow{\lambda} C'\} \\
\bar{S}_{i+1} =& \{C' \mid C' \in (S/\sim) \setminus \textstyle\bigcup_{j=0}^{i} \bar{S}_j, \\
& \qquad \exists C \in \bar{S}_i \cdot C \xrightarrow{a} C' \vee (st(C) \wedge C \xhookrightarrow{\lambda} C')\}.
\end{aligned}$$

We then have $\bar{S} = \bigcup_{i=0}^{\infty} \bar{S}_i$, $\bar{R}^I = \bigcup_{i=0}^{\infty} \bar{R}_i^I$, and $\bar{R}^M = \bigcup_{i=0}^{\infty} \bar{R}_i^M$. For the initial distribution we find for all equivalence classes $C \in \bar{S}$, that: $\bar{\alpha}(C) = \sum_{x \in C} \alpha(x)$. Note that we have $\bar{S} \subset S/\sim$.

The quotient under bisimulation is defined inductively to avoid the inclusion of unreachable equivalence classes. We will see (cq. Theorem 22) that this means that strong bisimulation quotients are unique up to isomorphism. If S is finite then the inductive definition of $[P]_\sim$ must terminate as the set of equivalence classes $S/\sim$ is finite.

We can lift the notion of strong bisimulation from states to I/O-IMCs by considering the *disjoint union* of two I/O-IMCs and showing that there exists a strong bisimulation on this union which relates the initial distributions of the two I/O-IMCs.

Definition 57. *Given two I/O-IMCs* $P = (S, A, R^I, R^M, \alpha)$ *and* $\bar{P} = (\bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ *with disjoint state spaces, identical input and output actions, the* disjoint union *of* P *and* $\bar{P}$*, written* $P \cup \bar{P}$ *is the I/O-IMC* $(\tilde{S}, \tilde{A}, \tilde{R}^I, \tilde{R}^M, \tilde{\alpha})$ *, where we have* $\tilde{S} = S \cup \bar{S}$*,* $\tilde{A}^O = A^O = \bar{A}^O$*,* $\tilde{A}^I = A^I = \bar{A}^I$*,* $\tilde{A}^H = A^H \cup \bar{A}^H$*,* $\tilde{R}^I = R^I \cup \bar{R}^I$*,* $\tilde{R}^M = R^M \cup \bar{R}^M$*, and for all states* $x \in \tilde{S}$

$$\tilde{\alpha}(x) = \begin{cases} 1/2 \cdot \alpha(x) & \text{, if } x \in S \\ 1/2 \cdot \bar{\alpha}(x) & \text{, if } x \in \bar{S}. \end{cases}$$

The disjoint union of two I/O-IMCs can be interpreted as the process that uniformly at random picks one of the I/O-IMCs and then behaves as this I/O-IMC. In practice, we will only use the union of two I/O-IMCs to lift the two bisimulations to I/O-IMCs. However, we first show that there is a close connection between strong bisimulations on the union of two I/O-IMCs and strong bisimulations on the I/O-IMCs themselves.

Theorem 18. *Given two I/O-IMCs* P *and* $\bar{P}$ *with disjoint state spaces, and identical input and output actions, two states* $x, y \in S$ *are strongly bisimilar in* P *if and only if they are strongly bisimilar in* $P \cup \bar{P}$*:*

$$x \sim_P y \Leftrightarrow x \sim_{P \cup \bar{P}} y.$$

Proof. We first prove the implication in the right direction. Since $x \sim_P y$ we must find a strong bisimulation $\mathcal{E}_1$ for P such that $x \mathcal{E}_1 y$. We define the relation $\mathcal{E}_2$ on $S \cup \bar{S}$ as follows:

$$\mathcal{E}_2 = \mathcal{E}_1 \cup \{(x', x') \mid x' \in \bar{S}\}.$$

It can be easily shown that $\mathcal{E}_2$ is a strong bisimulation on $P \cup \bar{P}$ and then we have $x \sim_{P \cup \bar{P}} y$.

We now prove the implication in the left direction. Since $x \sim_{P \cup \bar{P}} y$ we must find a strong bisimulation $\mathcal{E}_2$ for $P \cup \bar{P}$ such that $x \mathcal{E}_2 y$. We define the relation $\mathcal{E}_1$ on S as follows:

$$\mathcal{E}_1 = \{(x', y') \mid x', y' \in S \wedge x' \mathcal{E}_2 y'\}.$$

It can be easily shown that $\mathcal{E}_1$ is a strong bisimulation for P and then we have $x \sim_P y$. □

Theorem 18 has an important consequence for the strong-bisimilarity equivalence classes of I/O-IMCs and their unions. For two I/O-IMCs P and $\bar{P}$ with identical visible actions we have that a strong bisimilarity equivalence class C of their union $P \cup \bar{P}$ consists of exactly one strong bisimilarity equivalence class C_P of P and one equivalence class $C_{\bar{P}}$ of $\bar{P}$, i.e., $C = C_P \cup C_{\bar{P}}$.

We now lift strong bisimulation to I/O-IMCs.

Definition 58 (Strong bisimulation for I/O-IMCs)**.** *Given two I/O-IMCs* P *and* $\bar{P}$ *with disjoint state spaces and identical visible actions,* P *is strongly bisimilar to* $\bar{P}$*, written* $P \sim \bar{P}$*, if there exists a strong bisimulation* $\mathcal{E}$ *on their union* $P \cup \bar{P}$*, such that for all equivalence classes* C *of* $\mathcal{E}$ *we have:*

$$\alpha(C \cap S) = \alpha(C \cap \bar{S}).$$

Two I/O-IMCs P and $\bar{P}$ with non-disjoint state spaces S and $\bar{S}$ and identical visible actions are strongly bisimilar if we can find disjoint I/O-IMCs P' and $\bar{P}'$ such that $P \equiv P'$ and $\bar{P} \equiv \bar{P}'$ and P' and $\bar{P}'$ are strongly bisimilar.

In the following we assume that whenever we consider two I/O-IMCs their state spaces are disjoint. The following results also hold for I/O-IMCs whose state spaces are not disjoint if we consider isomorphic I/O-IMCs with disjoint state spaces as above.

First we establish that strong bisimilarity is indeed an equivalence relation for I/O-IMCs.

Theorem 19. *Strong bisimilarity is an equivalence relation. For I/O-IMCs P, $\bar{P}$, and $\tilde{P}$ with identical visible actions, we have that strong bisimilarity is*

1. *reflexive, i.e., $P \sim P$,*
2. *symmetric, i.e., $P \sim \bar{P} \implies \bar{P} \sim P$, and*
3. *transitive, i.e., $P \sim \bar{P} \wedge \bar{P} \sim \tilde{P} \implies P \sim \tilde{P}$.*

Proof. The proofs for reflexivity and symmetry of strong bisimilarity are trivial. We now show that strong bisimilarity is also transitive. For simplicity, we assume I/O-IMCs P, $\bar{P}$, and $\tilde{P}$ have disjoint state spaces. Since we have $P \sim \bar{P}$ and $\bar{P} \sim \tilde{P}$ we find strong bisimulations $\mathcal{E}_1$ on $P \cup \bar{P}$ and $\mathcal{E}_2$ on $\bar{P} \cup \tilde{P}$. We now define the relation $\mathcal{E}_3$ on $S \cup \tilde{S}$ as follows:

$$\begin{aligned}\mathcal{E}_3 =& \{(x,z) \mid x \in S, z \in \tilde{S}, \exists y \in \bar{S} \cdot x\mathcal{E}_1 y \mathcal{E}_2 z\} \cup \\ & \{(z,x) \mid x \in S, z \in \tilde{S}, \exists y \in \bar{S} \cdot x\mathcal{E}_1 y \mathcal{E}_2 z\} \cup \\ & \{(x,x') \mid x, x' \in S, x\mathcal{E}_1 x'\} \cup \{(z,z') \mid z, z' \in \tilde{S}, z\mathcal{E}_2 z'\}.\end{aligned}$$

It can now be easily shown that $\mathcal{E}_3$ is a strong bisimulation using the fact that $\mathcal{E}_1$ and $\mathcal{E}_2$ are strong bisimulations. □

As we might expect, strong bisimulation is a strictly weaker equivalence than isomorphism.

Theorem 20. *Isomorphism implies strong bisimulation. For I/O-IMCs P and $\bar{P}$ with identical visible actions we have:*

$$P \equiv \bar{P} \implies P \sim \bar{P},$$

but the reverse does not hold.

Proof. Given that $P \equiv \bar{P}$ we find a bijection f from S to $\bar{S}$ that satisfies the conditions of Definition 54. It is easy to show that the reflexive closure of $\{(x, f(x) \mid x \in S\}$ is a strong bisimulation on the union of P and $\bar{P}$. A counterexample for the implication $P \sim \bar{P} \implies P \equiv \bar{P}$ is also easy to construct, for instance by using the fact that a transitions with a hidden action can be simulated (for strong bisimulation) by a transition with a different hidden action. □

An I/O-IMC is strongly bisimilar to its strong bisimulation quotient.

Theorem 21. *Given an I/O-IMC P, we have $P \sim [P]_{\sim}$.*

Proof. Standard. □

Finally, we wish to show that the strong bisimilarity of two I/O-IMCs is equivalent to the isomorphism of their strong bisimulation quotients. To prove this we need the following lemma that shows that for two strongly bisimilar I/O-IMCs strong bisimilarity on their union relates their initial distributions. This lemma makes it easier to relate the equivalence classes of the union of two strongly bisimilar I/O-IMCs to the equivalence classes of the I/O-IMCs themselves.

Lemma 10. *Given two I/O-IMCs P and $\bar{P}$ with identical visible actions, if P and $\bar{P}$ are strongly bisimilar then strong bisimilarity on $P \cup \bar{P}$ preserves the initial distributions of P and $\bar{P}$. For each equivalence class C of $(S \cup \bar{S})/\sim_{P\cup\bar{P}}$ we have:*

$$\tilde{\alpha}(C \cap S) = \tilde{\alpha}(C \cap \bar{S}),$$

where $\tilde{\alpha}$ is the initial distribution of $P \cup \bar{P}$.

Proof. From the bisimilarity of P and $\bar{P}$ we know that there exists a strong bisimulation $\mathcal{E}$ on $P \cup \bar{P}$ that preserves the initial distributions α and $\bar{\alpha}$ of P and $\bar{P}$, respectively. Now, since $\sim_{P\cup\bar{P}}$ is the largest strong bisimulation on $P \cup \bar{P}$ we have for any two states $x, y \in S \cup \bar{S}$ that:

$$x \mathcal{E} y \implies x \sim_{P\cup\bar{P}} y.$$

Since $\mathcal{E}$ and $\sim_{P\cup\bar{P}}$ are equivalence relations it follows that for any equivalence class C of $\sim_{P\cup\bar{P}}$ we can find a countable set of equivalence classes $\mathcal{D}$ of $\mathcal{E}$ such that $C = \bigcup_{D\in\mathcal{D}} D$. Now we find:

$$\tilde{\alpha}(C \cap S) = \sum_{D\in\mathcal{D}} \tilde{\alpha}(D \cap S) = \sum_{D\in\mathcal{D}} \tilde{\alpha}(D \cap \bar{S}) = \tilde{\alpha}(C \cap \bar{S}).$$

□

We now show that strong bisimilarity implies the isomorphism of strong bisimulation quotients.

Lemma 11. *Given I/O-IMCs P and $\bar{P}$, if P is strongly bisimilar to $\bar{P}$ then $[P]_{\sim}$ is isomorphic to $\left[\bar{P}\right]_{\sim}$.*

Proof. We prove Lemma 11 by constructing a bijection from the equivalence classes of $[P]_{\sim}$ to the equivalence classes of $\left[\bar{P}\right]_{\sim}$ and showing that it is indeed an isomorphism.

Since P and $\bar{P}$ are strongly bisimilar, we have from Lemma 10, that strong bisimilarity on $P \cup \bar{P}$ (i.e., $\sim_{P\cup\bar{P}}$) preserves the initial distributions of P and $\bar{P}$. Furthermore, from Theorem 18 it follows that any equivalence class C in $(S \cup \bar{S})/\sim_{P\cup\bar{P}}$ consists of exactly one equivalence class $C_P \in S/\sim_P$ of P and one equivalence class $C_{\bar{P}} \in \bar{S}/\sim_{\bar{P}}$ of $\bar{P}$. We now construct the function f from the equivalence classes S' of $[P]_{\sim}$ to the

equivalence classes $\bar{S}'$ of $\left[\bar{P}\right]_\sim$. For any equivalence class C_P in S' let $C \in (S \cup \bar{S})/\sim_{P \cup \bar{P}}$ and $C_{\bar{P}} \in \bar{S}/\sim_{\bar{P}}$ be such that $C = C_P \cup C_{\bar{P}}$. We now find:

$$f(C_P) = \begin{cases} C_{\bar{P}} & \text{, if } C_{\bar{P}} \in \bar{S}' \\ \text{undefined} & \text{, otherwise.} \end{cases}$$

We wish to show that f is an isomorphism from S' to $\bar{S}'$. To do this we must first show that f is a bijection, i.e.,

- f is a total function: $\forall C_P \in S' \cdot f(C_P)$ is defined,
- f is injective: $\forall C_P, C'_P \in S' \cdot f(C_P) = f(C'_P) \implies C_P = C'_P$, and
- f is surjective: $\forall C_{\bar{P}} \in \bar{S}' \cdot \exists C_P \cdot f(C_P) = C_{\bar{P}}$.

For an equivalence class C_P in S' we find some index $i \in \mathbb{N}$ such that $C_P \in S'_i$. Let C and $C_{\bar{P}}$ again be equivalence classes of $P \cup \bar{P}$ and $\bar{P}$ respectively such that $C = C_P \cup C_{\bar{P}}$. We prove by induction on i that $C_{\bar{P}}$ is also in $\bar{S}'$, using the following induction assumption:

$$\forall C'_P \in S' \cdot C'_P \in S'_j \wedge j < i \implies \exists C'_{\bar{P}} \in S'_{\bar{P}} \cdot f(C'_P) = C'_{\bar{P}}. \qquad (5.4)$$

For the case that i equals zero we find, by Definition 56, that $\alpha(C_P) > 0$. Since P and $\bar{P}$ are strongly bisimilar we find:

$$\bar{\alpha}(C_Q) = \bar{\alpha}(C \cap \bar{S}) = \alpha(C \cap S) = \alpha(C_P).$$

This means C_Q is also in $\bar{S}'$.

For the case that i is larger than zero, we find, by Definition 56, that there exists an equivalence class C'_P in S'_{i-1} such that there is a strong transition from C'_P to C_P. Let C' and $C'_{\bar{P}}$ be equivalence classes of $P \cup \bar{P}$, respectively $\bar{P}$, such that $C' = C'_P \cup C'_{\bar{P}}$. By (5.4), we have that $C'_{\bar{P}}$ is in $\bar{S}'$. It is now easy to show from the strong bisimilarity of states in C'_P and $C'_{\bar{P}}$ that we must find an equivalence class $C''_{\bar{P}}$ of $\bar{P}$ such that there is a matching transition from $C'_{\bar{P}}$ to $C''_{\bar{P}}$ where the states of $C''_{\bar{P}}$ are strongly bisimilar to the states of C_P. It follows that $C''_{\bar{P}}$ is in fact $C_{\bar{P}}$ and $C_{\bar{P}}$ is in $\bar{S}'$.

We have shown that f is a total function from S' to $\bar{S}'$. Injectivity and surjectivity now follow directly from the fact that for each equivalence class $C_P \in S'$ we find an equivalence class C of $P \cup \bar{P}$ such that $C = C_P \cup f(C_P)$.

It remains to show that f is an isomorphism. For an arbitrary equivalence class C_P of P we find that:

- For the initial distributions we have $\alpha'(C_P) = \bar{\alpha}'(f(C_P))$, since strong bisimilarity of P and $\bar{P}$ gives us that $\alpha(C_P) = \bar{\alpha}(f(C_P))$.
- For any visible transition $C_P \xrightarrow{a} C'_P$ we find, since $P \sim \bar{P}$, that there exists a transition $f(C_P) \xrightarrow{a} C'_{\bar{P}}$ where the states of C'_P are strongly bisimilar to the states in $C'_{\bar{P}}$ for $P \cup \bar{P}$. It must then be the case that $C'_{\bar{P}} = f(C'_P)$.

- Similarly we find that hidden transitions and Markovian transitions of C_P must be matched by $f(C_P)$. Note that for both C_P and $f(C_P)$ hidden transitions will always be labelled τ.
- For $st(C_P)$ we have that all states in C_P and $f(C_P)$ are pairwise equivalent according to $=_s$. Then, C_P and $f(C_P)$ are also equivalent according to $=_s$.

□

Lemma 12. *Given I/O-IMCs P and $\bar{P}$, if $[P]_\sim$ is isomorphic to $\left[\bar{P}\right]_\sim$ then P is strongly bisimilar to $\bar{P}$.*

Proof. To prove Lemma 12 we will find a strong bisimulation on the states of $P \cup Q$ that relates their initial distributions.

Since P is strongly bisimilar to $[P]_\sim$ we find a strong bisimulation $\mathcal{E}_1$ on $P \cup [P]_\sim$ that relates their initial distributions. Similarly we find a strong bisimulation $\mathcal{E}_2$ on $\bar{P} \cup \left[\bar{P}\right]_\sim$ that relates the initial distributions of $\bar{P}$ and $\left[\bar{P}\right]_\sim$. Additionally, we find an isomorphism f from the equivalence classes of $[P]_\sim$ to the equivalence classes of $\left[\bar{P}\right]_\sim$. We use these relations to define the equivalence relation $\mathcal{E}_3$ on $S \cup \bar{S}$:

$$\begin{aligned}\mathcal{E}_3 =&\{(x,y) \mid x \in S, y \in \bar{S}, \exists C \in S' \cdot x\mathcal{E}_1 C \wedge f(C)\mathcal{E}_2 y\} \cup \\ &\{(y,x) \mid x \in S, y \in \bar{S}, \exists C \in S' \cdot x\mathcal{E}_1 C \wedge f(C)\mathcal{E}_2 y\} \cup \\ &\{(x,x') \mid x, x' \in S, x\mathcal{E}_1 x'\} \cup \{(y,y') \mid y, y' \in \bar{S}, y\mathcal{E}_2 y'\}.\end{aligned}$$

It can easily be shown that $\mathcal{E}_3$ is a strong bisimulation for $P \cup \bar{P}$ that relates the initial distributions of P and $\bar{P}$. □

Theorem 22. *Given I/O-IMCs P and $\bar{P}$, P is strongly bisimilar to $\bar{P}$ if and only if $[P]_\sim$ is isomorphic to $\left[\bar{P}\right]_\sim$.*

Proof. Theorem 22 follows directly from Lemmas 11 and 12. □

Finally, we look at whether or not strong bisimilarity is a cogruence with respect to parallel composition and hiding.

Theorem 23. *Strong bisimilarity is a congruence with respect to parallel composition and hiding. I.e., for I/O-IMCs P, $\bar{P}$, and $\tilde{P}$, such that P and $\bar{P}$ have identical visible actions and are both compatible with $\tilde{P}$ we have*

$$P \sim \bar{P} \implies P \| \tilde{P} \sim \bar{P} \| \tilde{P},$$

and

$$P \sim \bar{P} \implies P \backslash B \sim \bar{P} \backslash B,$$

for any set of output actions $B \subset A^O$.

Proof. Standard. □

Although strong bisimulation fulfills all our criteria for equivalence relations on I/O-IMCs, we will in the next subsection we will consider one more equivalence that is even weaker, but still fulfills our criteria.

5.5.3 Weak Bisimulation

As our most important notion of equivalence we use the notion of *weak bisimulation*. This equivalence abstracts from internal transitions (interactive as well as Markovian) and cumulates Markovian transition rates. It also uses the maximal progress assumption to abstract away from Markovian transitions that are taken with probability zero and the identity (according to $=_s$) of states that are occupied with probability zero. We use weak bisimulation as our main equivalence relation as it is a congruence with respect to parallel composition and hiding, can be computed efficiently [23], preserves the stochastic properties of I/O-IMCs (as we will see in Subsection 7.2), and has been shown to be effective in practice [6, 13].

Definition 59 (Weak bisimulation for states)**.** *Given I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, an equivalence relation $\mathcal{E}$ on S is a* weak bisimulation *if and only if for all pairs of states $x, y \in S$ such that $x\mathcal{E}y$ we have:*

1. *All directly observable, weak, and interactive transitions emerging from x can be simulated by y:*

$$\forall a \in A^O \cup A^I \cdot \left(\forall x' \in S \cdot x \xrightarrow{a}\!\!\!\!\rightarrow x' \implies \exists y' \in S \cdot y \xrightarrow{a}\!\!\!\!\rightarrow y' \wedge x'\mathcal{E}y'\right).$$

2. *All internal, weak, and interactive transitions emerging from x and going to a different equivalence class can be simulated by y:*

$$\forall x' \in S \cdot x \longrightarrow\!\!\!\!\rightarrow x' \wedge \neg(x\mathcal{E}x') \implies \exists y' \in S \cdot y \longrightarrow\!\!\!\!\rightarrow y' \wedge x'\mathcal{E}y'.$$

3. *If x is stable, then y must be able to reach, using an internal weak transition, a stable state y'. Furthermore y' must be able to simulate all Markovian transitions emerging from x and going to a different equivalence class:*

$$\begin{aligned} st(x) \implies & \exists y' \in S \cdot y \longrightarrow\!\!\!\!\rightarrow y' \wedge st(y') \wedge \\ & \forall C \in S/\mathcal{E} \cdot x \notin C \wedge x \hookrightarrow\!\!\!\!\!\rightarrow C \implies y' \hookrightarrow\!\!\!\!\!\rightarrow C \end{aligned}$$

4. *If x is stable, then y must be able to make an internal, weak, transition to a stable state y' that is identical to x according to $=_s$:*

$$st(x) \implies \exists y' \in S \cdot y \longrightarrow\!\!\!\!\rightarrow y' \wedge st(y') \wedge y' =_s x.$$

Two states x and y are weakly bisimilar for P, written $x \approx_P y$ if there exists a weak bisimulation for P that relates x and y. We leave out the subscript when clear from context.

It is important to note that the third property of weak bisimulation does not just concern Markovian transitions, but also internal weak transitions to stable states. For two weakly bisimilar states x and y we have: $st(x) \implies y \longrightarrow\!\!\!\!\rightarrow y' \wedge st(y')$ even if x has no outgoing Markovian transitions. This singles out stable states as being fundamentally

different from unstable states. Consider for instance two states x and y where x has no outgoing transitions and y has only the internal transition $y \xrightarrow{\tau_i} y$. These states have exactly the same outgoing weak transitions ($x \longrightarrow\!\!\!\!\!\rightarrow x$ and $y \longrightarrow\!\!\!\!\!\rightarrow y$), but they are not weakly bisimilar, because y cannot reach a stable state, while x can.

Theorem 24. *Given an I/O-IMC P, weak bisimilarity on P is the largest weak bisimulation on P.*

Proof. Standard. □

We also define the quotient under weak bisimulation.

Definition 60. *Given an I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, its* quotient under weak bisimulation *is an I/O-IMC $[P]_{\approx} = (S', \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ with $\bar{A} = (A^I, A^O, \{\tau\})$ where states and transitions are defined inductively as follows for $i \in \mathbb{N}$:*

$$
\begin{aligned}
S'_0 =& \{C \mid C \in S/\approx, \alpha(C) > 0\} \\
\bar{R}^I_i =& \{(C, a, C') \mid C \in S'_i, C' \in S/\approx, a \in A^I \cup A^O, \exists x \in C, y \in C' \cdot x \xrightarrow{a}\!\!\!\!\!\rightarrow y\} \cup \\
& \{(C, \tau, C') \mid C \in S'_i, C' \in S/\approx, \exists x \in C, y \in C' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge C \neq C'\} \cup \\
& \{(C, \tau, C) \mid C \in S'_i, \forall x \in C \cdot \\
& \qquad (\nexists C'' \in S/\approx, y \in C'', a \in A^O_P \cdot x \xrightarrow{a}\!\!\!\!\!\rightarrow y) \wedge \\
& \qquad (\nexists C'' \in S/\approx, y \in C'' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge C \neq C'') \wedge \\
& \qquad (\nexists C'' \in S/\approx, y \in C'' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge st(y))\} \\
\bar{R}^M_i =& \{(C, \lambda, C') \mid C \in S'_i, C' \in S/\approx, C \neq C', \exists x \in C \cdot st(x) \wedge x \xrightarrow{\lambda} C'\} \\
S'_{i+1} =& \{C' \mid C' \in (S/\approx) \setminus \textstyle\bigcup_{j=0}^{i} S'_j, \exists C \in S'_i \cdot C \xrightarrow{a} C' \vee C \xrightarrow{\lambda} C'\}.
\end{aligned}
$$

We then have $S' = \bigcup_{i=0}^{\infty} S'_i$, $\bar{R}^I = \bigcup_{i=0}^{\infty} \bar{R}^I_i$, and $\bar{R}^M = \bigcup_{i=0}^{\infty} \bar{R}^M_i$. For the initial distribution we find for all equivalence classes $C \in S'_0$, that: $\bar{\alpha}(C) = \sum_{x \in C} \alpha(x)$. Note that we have $S' \subset S/\mathcal{E}$.

There is one important aspect of the weak bisimulation quotient that does not immediately follow from the definition of the strong bisimulation quotient. This has to do with the presence of internal self-loops on the equivalence classes. First of all, it is obvious that for any state x in an I/O-IMC we have $x \longrightarrow\!\!\!\!\!\rightarrow x$. Then the question arises: when should such an internal self-loop appear in the quotient model? One choice would be to add an internal self-loop to each equivalence class in the quotient. However, we have seen already that a state with an internal self-loop can be fundamentally different from a state without one. This also means that adding no internal self-loops to any equivalence class of a weak bisimulation quotient is equally undesirable.

To solve this dilemma, we simply apply our fairness assumptions (see Section 4.4). If the states in a particular equivalence class can *fairly* generate unboundedly many internal events, then we add an internal self-loop to this class. Now, we know from our discussion of fairness that such an unbounded sequence of internal events is only allowed for states that are *divergent*, i.e., states that cannot reach a stable state. From the third

clause of weak bisimulation it follows that divergent states can only be weakly bisimilar to other divergent states. The consequence for the weak bisimulation quotient of an I/O-IMC is that each equivalence class consists either of only divergent states (in this case the equivalence class will have an internal self-loop) or no divergent states (in this case the equivalence class will not have an internal self-loop).

As for strong bisimulation we can relate weak bisimilarity on the union of two I/O-IMCs to weak bisimilarity on the individual I/O-IMCs.

Theorem 25. *Given two I/O-IMCs P and $\bar{P}$ with disjoint state spaces, identical input and output actions, two states $x, y \in S$ are weakly bisimilar in P if and only if they are weakly bisimilar in $P \cup \bar{P}$:*

$$x \approx_P y \Leftrightarrow x \approx_{P \cup \bar{P}} y.$$

Proof. Similar to the proof of Theorem 18. □

Weak bisimulation can be lifted to I/O-IMCs in the same way strong bisimulation was lifted to I/O-IMCs.

Definition 61 (Weak Bisimulation for I/O-IMCs)**.** *Given the following two I/O-IMCs $P = (S, A, R^I, R^M, \alpha)$ and $\bar{P} = (\bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ with disjoint state spaces, we say that P is weakly bisimilar to $\bar{P}$ if $A^O = \bar{A}^O$ and $A^I = \bar{A}^I$ and we can find a weak bisimulation $\mathcal{E}$ on $P \cup \bar{P}$, such that for all equivalence classes C of $\mathcal{E}$ we have:*

$$\alpha_P(C \cap S) = \bar{\alpha}(C \cap \bar{S}).$$

We write $P \approx \bar{P}$. We say that two I/O-IMCs with non-disjoint state spaces are weakly bisimilar if we can find two isomorphic I/O-IMCs with disjoint state spaces that are weakly bisimilar.

As we found for strong bisimilarity we have that weak bisimilarity is an equivalence relation.

Theorem 26. *Weak bisimilarity is an equivalence relation.*

Proof. Follows the proof for strong bisimilarity. □

Weak bisimilarity, however, is a strictly weaker equivalence than strong bisimilarity.

Theorem 27. *Given two I/O-IMCs P and $\bar{P}$ with identical visible actions we have:*

$$P \sim \bar{P} \implies P \approx \bar{P},$$

but the reverse does not hold.

Proof. It is easy to show that any strong bisimulation on an I/O-IMC is also a weak bisimulation on the same I/O-IMC. For an example that shows that the reverse does not hold consider I/O-IMCs P and $\bar{P}$ such that P has two states x and y and Q has one state z. The only transition in P is an internal transition from x to y, whereas $\bar{P}$ has no transitions. It is easy to see that P and $\bar{P}$ are weakly bisimilar but not strongly bisimilar. □

It follows of course that weak bisimilarity is strictly weaker than isomorphism. We now show the expected result that an I/O-IMC is weakly bisimilar to its weak bisimulation quotient. To do this we first establish that equivalence classes in the weak bisimulation quotient are stable if and only if they contain at least one stable state.

Lemma 13. *Given an I/O-IMC P, for any equivalence class C of $[P]_{\approx}$ we have:*

$$st(C) \Leftrightarrow \exists x \in C \cdot st(x).$$

Proof. We first prove the implication in the right direction. Assume $C \in S'$ is stable. It follows that C has no outgoing output transitions and C has no outgoing internal transitions to different equivalence classes in S'. We then have that no state x in C has outgoing weak output transitions or outgoing weak internal transitions to states in equivalence classes other than C. We also find that there is no internal self-loop $C \xrightarrow{\tau} C$. This means that the condition

$$\begin{aligned}\forall x \in C \cdot (&\nexists C' \in S/\approx, y \in C', a \in A_P^O \cdot x \xrightarrow{a}\!\!\!\!\rightarrow y) \wedge \\ &(\nexists C' \in S/\approx, y \in C' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge C \neq C') \wedge \\ &(\nexists C' \in S/\approx, y \in C' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge st(s'))\end{aligned}$$

does not hold for C. Since C has no visible transitions and no hidden transitions to other equivalence classes this must mean that there is a state x in C that can reach a stable state y via a weak internal transition. Now, if y does not lie in C then there must be an internal transition from C to the equivalence class of y, but this is a contradiction with the fact that C is stable. It follows that y is in C and then C indeed contains a stable state.

We prove the implication in the left direction by contradiction. Assume then that there is a stable state x in C, but C is not stable. Then, C must have an outgoing output transition or an outgoing internal transition. There are three possibilities:

1. There exists a state $y \in C$ and a state z in an equivalence class $C' \in S'$ and an action $a \in A_P^O$ such that $y \xrightarrow{a}\!\!\!\!\rightarrow z$,
2. There exists a state $y \in C$ and a state z in an equivalence class $C' \in S'$ such that $C \neq C'$ and $y \longrightarrow\!\!\!\!\!\rightarrow z$, or
3. We find an internal self-loop $C \xrightarrow{\tau} C$.

For the first two cases we find $x \approx_P y$ and then x must have a matching outgoing output or internal transition, which is a contradiction with the fact that x is stable. In the third case we find for x (since it is in C) that the condition

$$\begin{aligned}&(\nexists C' \in S/\approx, y \in C', a \in A_P^O \cdot x \xrightarrow{a}\!\!\!\!\rightarrow y) \wedge \\ &(\nexists C' \in S/\approx, y \in C' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge C \neq C') \wedge \\ &(\nexists C' \in S/\approx, y \in C' \cdot x \longrightarrow\!\!\!\!\!\rightarrow y \wedge st(y))\end{aligned}$$

holds. This means that it cannot reach a stable state with a weak internal transition. However, this is a direct contradiction with the fact that x is stable itself (recall: $x \longrightarrow\!\!\!\!\!\rightarrow x$). □

We now show that an I/O-IMC is weakly bisimilar to its weak bisimilarity quotient.

Theorem 28. *Given an I/O-IMC $P = (S, A, R^I, R^M, \alpha)$, we have: $P \approx [P]_{\approx}$.*

Proof. Let $\mathcal{E}$ be the following relation on $S \cup S'$:

$$\begin{aligned}\mathcal{E} =&\{(x, C) \mid x \in S, C \in S', x \in C\} \cup \{(C, x) \mid x \in S, C \in S', x \in C\} \cup \\ &\{(x, y) \mid x, y \in S, x \approx_P y\} \cup \{(C, C) \mid C \in S'\}\end{aligned}$$

Given Lemma 13 it is easy to show that $\mathcal{E}$ is indeed a weak bisimulation on $P \cup [P]_{\approx}$ that relates the initial distributions of P and $[P]_{\approx}$. □

Finally, we show that the weak bisimilarity of two I/O-IMCs is equivalent to the isomorphism of their weak bisimulation quotients. The proof follows the same lines as the proof of Theorem 22, except that we must be careful in the case of internal self-loops in the weak bisimulation quotients. The following lemma states that these self-loops do not cause a problem. We therefor first prove a lemma regarding the self-loops of the quotients of weakly bisimilar I/O-IMCs.

Lemma 14. *Given two weakly bisimilar I/O-IMCs P and $\bar{P}$ with identical visible actions, for equivalence classes C_P and $C_{\bar{P}}$ of the weak bisimulation quotients $[P]_{\approx}$ and $\left[\bar{P}\right]_{\approx}$ respectively, such that $C_P \approx_{[P]_{\approx} \cup \left[\bar{P}\right]_{\approx}} C_{\bar{P}}$ we find:*

$$C_P \xrightarrow{\tau} C_P \Leftrightarrow C_{\bar{P}} \xrightarrow{\tau} C_{\bar{P}}.$$

Proof. Since C_P is weakly bisimilar to $C_{\bar{P}}$ in the union $[P]_{\approx} \cup \left[\bar{P}\right]_{\approx}$ we have that any state $x \in C_P$ is weakly bisimilar to any state $y \in C_{\bar{P}}$ in the union $P \cup Q$. Assume now that there is an internal transition $C_P \xrightarrow{\tau} C_P$. We will show that it follows that there is also an internal transition $C_{\bar{P}} \xrightarrow{\tau} C_{\bar{P}}$ by contradiction. Assume then that there is no transition $C_{\bar{P}} \xrightarrow{\tau} C_{\bar{P}}$. We have that the condition

$$\begin{aligned}\forall x \in C_P \cdot (&\nexists C'_P \in S_P/\approx, y \in C'_P, a \in A^O_P \cdot x \xrightarrow{a} y) \wedge \\ &(\nexists C'_P \in S_P/\approx, y \in C'_P \cdot x \longrightarrow y \wedge C_P \neq C'_P) \wedge \\ &(\nexists C'_P \in S_P/\approx, y \in C'_P \cdot x \longrightarrow y \wedge st(s'))\end{aligned}$$

holds for C_P, but does not hold for $C_{\bar{P}}$. We then find for any state x in C_P that it does not have any of the above weak transitions. But at the same time we must find a state y in $C_{\bar{P}}$ that does have one of the above outgoing transitions. This is in contradiction with the fact that x must be weakly bisimilar to y. We find by a similar reasoning that if $C_{\bar{P}}$ has an internal self-loop then C_P has an internal self-loop. □

Theorem 29. *Given I/O-IMCs P and Q, P is weakly bisimilar to Q if and only if $[P]_{\approx}$ is isomorphic to $[Q]_{\approx}$.*

Proof. The proof of Theorem 29 proceeds in the same way as the proof of Theorem 22, with the addition that the presence of internal self-loops is preserved by Lemma 14. □

Theorem 30. *Weak bisimilarity is a congruence with respect to parallel composition and hiding. I.e., for I/O-IMCs P, $\bar{P}$, and $\tilde{P}$, such that P and $\bar{P}$ have identical visible actions and are both compatible with $\tilde{P}$ we have*

$$P \approx \bar{P} \implies P \| \tilde{P} \approx \bar{P} \| \tilde{P},$$

and

$$P \approx \bar{P} \implies P \backslash B \approx \bar{P} \backslash B,$$

for any set of output actions $B \subset A^O$.

Proof. Standard. □

5.6 Stochastic reachability

Reachability has been studied extensively in computer science. Intuitively, we say a configuration of a system is reachable if it *can* be reached. For non-deterministic models, such as the IOA we studied in Chapter 4, this means that a state is reachable if there is some resolution of the non-determinism with which the state is eventually reached. For stochastic models we often talk about the reachability probability of a state. This is the probability of eventually reaching the state in question when we execute the model. However, it is also common to say that a state is reachable in a stochastic model if its reachability probability is strictly greater than zero. In this thesis, we are dealing with models that are both non-deterministic and stochastic and we will say a state is reachable if there is a resolution of the non-determinism such that the probability to reach that state is strictly greater than zero.

For an IOA, we have seen that a state x is reachable if there is a fair path from the initial state to x (see Definition 29). In our setting, this is unfortunately not the case. Due to the maximal progress assumption, we have that Markovian transitions that emanate from unstable states will be taken with probability zero. We therefore first identify a subset of the transitions that *may* be taken with probability greater than zero. We call these transitions *plausible*. Differently put, transitions that are *not* plausible are never taken.

Definition 62 (Plausible Transition)**.** *Given a state x in an I/O-IMC P, a transition in P starting at x is called* plausible *if it is either an interactive transition, or x is stable and the transition is Markovian.*

We now extend the usual definition of paths to *plausible* paths in the same way. A path is plausible when it consists of plausible transitions.

Definition 63 (Plausible Path)**.** *Given a state x in an I/O-IMC P, a finite path σ in P emanating from x is called* plausible *if it consists of plausible transitions. In other words we can say: if there is a series of states $x_1, \ldots, x_n$ such that $x_1 = x$ and for $1 \leq i < n$,*

$$\left(\exists a \in A_P \wedge x_i \xrightarrow{a} x_{i+1}\right) \vee \left(st(x_i) \wedge \exists \lambda \in \mathbb{R}_{>0} \cdot x_i \overset{\lambda}{\hookrightarrow} x_{i+1}\right),$$

then there is a plausible path from x to x_n.

It is useful to note here that we consider only finitely-branching I/O-IMCs, because otherwise even plausible transitions may be taken with probability zero.

Now we are ready to define reachability for I/O-IMCs in terms of plausible paths. Since a state is usually considered to be reachable if there is a path to that state, we instead introduce *stochastic* reachability. A state is stochastically reachable if it can be reached via a path, and this path has probability greater than zero to be taken for some resolution of the non-determinism.

Definition 64 (Stochastic Reachability)**.** *Given an I/O-IMC P with state space S, a state $y \in S$ is stochastically reachable in P, written $SR_P(y)$ if and only if there exists a state $x \in S$ with non-zero initial probability (i.e., $\alpha(x) > 0$) such that there is a plausible path from x to y. We write $SR(y)$ when the identity of the I/O-IMC is clear from context.*

Note that in the case $\alpha(x) > 0$ we find that x is stochastically reachable, because the empty path consisting only of state x itself is plausible by definition. Our intuition is that a state in an I/O-IMC is stochastically reachable if and only if there is some way to resolve the non-determinism such that the probability to reach that state is greater than zero for all time-points greater than zero. In Chapter 7 we will prove that this intuition is indeed correct for closed I/O-IMC.

But what about open I/O-IMCs? For an open I/O-IMC P we want that a state x is stochastically reachable if and only if we can find a compatible I/O-IMC P' such that $P \| P'$ is closed and there exists some state x' in P' such that $x \| x'$ is stochastically reachable for the closed I/O-IMC $P \| P'$.

Theorem 31. *Given an I/O-IMC P with state space S and actions A, a state $x \in S$ is stochastically reachable if and only if there exists a compatible I/O-IMC $\bar{P}$ with state space $\bar{S}$, such that $P \| \bar{P}$ is closed and there exists a state y in $\bar{S}$ such that $x \| y$ is stochastically reachable in $P \| \bar{P}$.*

Proof. We sketch the proof of Theorem 31. Given that the state x is stochastically reachable in P there must exists a plausible path σ from an initial state of P to x. We now construct a path $\bar{\sigma}$ in $\bar{P}$ such that the "parallel composition" of σ and $\bar{\sigma}$ is plausible in $P \| \bar{P}$. Each interactive transition in σ is matched by a similarly-labelled transition in $\bar{\sigma}$. Each Markovian transition in σ is matched by a Markovian transition in $\bar{\sigma}$. We can then pick $\bar{P}$ to be the I/O-IMC that consists exactly of the path $\bar{\sigma}$ where $\bar{\alpha}(\mathit{first}(\bar{\sigma})) = 1$.

It remains to show that if there exists a stochastically reachable state $x \| y$ in $P \| \bar{P}$, then x is stochastically reachable in P. We will show that this is the case in the proof of Lemma 15. □

In the remainder of this section we will study how stochastic reachability interacts with the architectural operations on I/O-IMCs.

5.6.1 Bisimulation and Stochastic Reachability

The bisimulation relations considered in this thesis are so-called *forward* bisimulation. Two states are bisimilar if they can simulate each other's *outgoing* transitions. This is

in contrast with so-called *backward* bisimulations, which equate states that can simulate each other's *incoming* transitions. Stochastic reachability is, in a sense, a "backward" property. A state is stochastically reachability if it has an *incoming* plausible path from an initial state. It is then no suprise that stochastic reachability is not preserved by strong or weak bisimulation.

Example 20. *Consider an I/O-IMC P with state space $S = \{x, y\}$, no actions or transitions, and an initial distribution that assigns probability one to state x. We now see that states x and y are both strongly and weakly bisimilar, but although x is stochastically reachable, y is not.*

Despite the fact that the bisimulations on states do not preserve stochastic reachability, we find that the bisimulations on I/O-IMCs (instead of on states) do, in a way, preserve stochastic reachability.

Theorem 32. *Given weakly bisimilar I/O-IMCs P and P', for any state x of P that is stochastically reachable we find a weakly bisimilar state x' of P' that is also stochastically reachable:*

$$\forall x \in S \cdot SR_P(x) \implies \exists x' \in S' \cdot x \approx_{P \cup P'} x' \wedge SR_{P'}(x'). \tag{5.5}$$

Proof. Let P and P' be weakly bisimilar I/O-IMCs as above and let x be a stochastically reachable state in P. By the definition of stochastic reachability we now have that there is a finite plausible path σ from an initial state in P to x. Let n denote the length of the path σ. We now prove Theorem 32 by induction on n.

We first consider the case that the length of σ is zero. Then we have that σ consists only of the state x. It immediately follows that x is an initial state of P, i.e., $\alpha_P(x) > 0$. Since weak bisimilarity on the union $P \cup P'$ must preserve initial distributions we find that there must exists an initial state x' of P' which is weakly bisimilar to x. Since x' is an initial state it is also stochastically reachable.

We now consider the case that the length of σ is greater than zero. We use the following induction assumption: for any plausible path σ_1 in P we find that:

$$\begin{aligned} &|\sigma_1| < |\sigma| \implies \\ &\quad \left(SR_P(last(\sigma_1)) \implies \exists x_1' \in S' \cdot x_1' \approx_{P \cup P'} last(\sigma_1) \wedge SR_{P'}(x_1')\right). \end{aligned} \tag{5.6}$$

We now show that it follows from (5.6) that (5.5) holds in the case that σ has length greater than zero. We have that σ must contain at least one transition and we consider the different possibilities for the last transition of σ. Since σ is plausible this transition must either be interactive or Markovian, but starting in a stable state.

Case: last transition is visible. We first consider the case that the last transiton of σ is an interactive transition labelled with a visible action. We then find a path σ_1, a state x_1 in S, and an action a in A_P^V such that $\sigma = \sigma_1 \circ x_1 \xrightarrow{a} x$. Trivially, σ_1 is also plausible and x_1 is stochastically reachable. Furthermore, we have that the length of σ_1 is $n - 1$. By the induction assumption we then find that there exists a state x_1' in

S' such that x_1 is weakly bisimilar to x_1' and x_1' is stochastically reachable in P'. We now have that x_1' must be able to simulate the (weak) transition $x_1 \stackrel{a}{\twoheadrightarrow} x$. We then find a state x' in S' such that there is a transition $x_1' \stackrel{a}{\twoheadrightarrow} x'$, with $x \approx x'$. Since x_1' is stochastically reachable and there is an interactive path from x_1' to x', it follows that x' is also stochastically reachable. We have then shown that (5.5) holds for the path σ.

Case: last transition is internal. We now consider the case that the last transition of σ is an interactive transition labelled with an internal action. We then find a path σ_1, a state x_1 in S and an action b in A^H such that $\sigma = \sigma_1 \circ x_1 \xrightarrow{b} x$. Again we find that σ_1 is plausible, x_1 is stochastically reachable, and the length of σ_1 is $n-1$. Applying (5.6), we find that there exists a state x_1' in S' which is stochastically reachable in P' and weakly bisimilar to x_1. Now we have two possibilities: either x_1 is weakly bisimilar to x or not. For the former we find – by transitivity – that x is also weakly bisimilar to x_1' and then (5.5) holds for the path σ. In the latter case, that x_1 is not weakly bisimilar to x we find that x_1' must simulate the weak transition $x_1 \twoheadrightarrow x$. This means there exists a state x' in S' such that $x_1' \twoheadrightarrow x'$ and $x \approx x'$. Since x_1' is stochastically reachable and there must be an interactive path from x_1' to x', it follows that x' is also stochastically reachable. We have again shown that (5.5) holds for the path σ.

Case: last transition is Markovian. We finally consider the case that the last transition of σ is a Markovian transition. We then find a path σ_1, a stable state x_1 in S and a rate $\lambda \in \mathbb{R}_{\geq 0}$ such that $\sigma = \sigma_1 \circ x_1 \xhookrightarrow{\lambda} x$. Once more we have that σ_1 is plausible, x_1 is stochastically reachable, and the length of σ_1 is $n-1$. From (5.6) we again have that there exists a state x_1' in S' such that x_1' is weakly bisimilar to x_1 and stochastically reachable in P'. We now consider whether x_1 is weakly bisimilar to x. In the case that they are, we have that x_1' is also weakly bisimilar to x, thus showing that (5.5) holds for the path σ.

If x_1 is not weakly bisimilar to x, then we have that x_1' must be able to internally reach a stable state x_1'' that simulate the Markovian transitions of x_1. Since x_1 moves to the equivalence class of $P \cup Q$ containing x with a rate greater than zero, we have that x_1'' does the same. There must then exist a state x' such that $x_1' \twoheadrightarrow x_1'' \xhookrightarrow{\mu} x'$ with $\mu \in \mathbb{R}_{\geq 0}$ and $x' \approx x$. From the fact that x_1' is stochastically reachable and x_1'' is stable we have that x' is also stochastically reachable in P', which means (5.5) holds. □

5.6.2 Parallel Composition and Stochastic Reachability

We now consider the relationship between parallel composition and stochastic reachability.

Example 21. *Consider two I/O-IMCs P and $\bar{P}$ such that P has an input action a and an output action b, while a is an output action and b is an input action for $\bar{P}$, let P and $\bar{P}$ both have three states x, y, and z respectively x', y', and z'. Finally we find transitions $x \xrightarrow{a} y \xrightarrow{b} z$ in P and $x' \xrightarrow{b} y' \xrightarrow{a} z'$ in $\bar{P}$. We also have self-loops $y \xrightarrow{a} y$, $z \xrightarrow{a} z$, $y' \xrightarrow{b} y'$, and $z' \xrightarrow{b} z'$ to ensure input-enabledness. Both I/O-IMCs have no Markovian transitions and the initial distributions assign probability one to x respectively x'.*

We now have that z is stochastically reachable in P and z' is stochastically reachable in $\bar{P}$, but the state $z\|z'$ is not stochastically reachable in $P\|\bar{P}$. In fact, the I/O-IMC $P\|\bar{P}$ has no transitions at all and only initial state $x\|x'$ is stochastically reachable.

From the above example we see that stochastic reachability is not preserved by parallel composition. This is to be expected as the intuition of stochastic reachability for an open I/O-IMC is that there exists *some* way of composing the I/O-IMC that achieves stochastic reachability, not that *all* compositions achieve this goal.

However, we can show that the reverse property holds. Namely, that stochastic reachability in a parallel composition is preserved when we "decompose" the parallel composition to its constituent I/O-IMCs. This is a consequence of the following lemma that tells us that the projections of a plausible paths are themselves plausible. We define the projection of composite paths onto component I/O-IMCs in the same way as we did for executions of IOA in Definition 32.

Lemma 15. *Given two compatible I/O-IMCs P and P' and a plausible path σ in $P\|P'$, the projections $\sigma\downarrow P$ and $\sigma\downarrow P'$ are plausible paths in P and P', respectively.*

Proof. We sketch the proof of Lemma 15. First, recall that if a state $x\|x'$ is stable in $P\|P'$ then the constituent states x and x' are stable in P respectively P'. It is now easy to prove the lemma by induction on the length of the path σ. □

Theorem 33. *Given compatible I/O-IMCs P and P', we find that if a state $x\|x'$ of $P\|P'$ is stochastically reachable then the states x and x' must be stochastically reachable in P and P' respectively:*

$$SR_{P\|Q}(x\|x') \implies SR_P(x) \wedge SR_Q(x').$$

Proof. Theorem 33 follows easily from Lemma 15 and the fact that if a state $x\|x'$ has non-zero initial probability in $P\|P'$ then both states x and x' have non-zero initial probability in P respectively P'. □

The above results can easily be extended to parallel compositions of more than two I/O-IMCs.

5.6.3 Hiding and Stochastic Reachability

Hiding does not affect the transitions in an I/O-IMC, but it does affect the role actions play in an I/O-IMC. By hiding actions we can turn output actions into internal actions. Note that hiding input actions is, by definition, not possible. Since hiding does not affect the transition relations, the only way hiding can influence stochastic reachability is to impact the stability of states. However, it is easy to see that hiding has no such effect, since any state that is stable has no outgoing output actions and thus is not affected by hiding. Conversely, any state that is unstable, will stay unstable after hiding actions since both output and internal transitions are assumed to occur instantaneously.

Theorem 34. *Given an I/O-IMC P and a set of actions B such that $B \subset A^O$, any state x is stochastically reachable in P if and only if x is stochastically reachable in $P\backslash B$:*

$$\forall x \in S_P \cdot SR_P(x) \Leftrightarrow SR_{P\backslash B}(x). \tag{5.7}$$

Proof. Straightforward. □

Figure 5.4 summarizes the modularity results for stability, and stochastic reachability of I/O-IMCs.

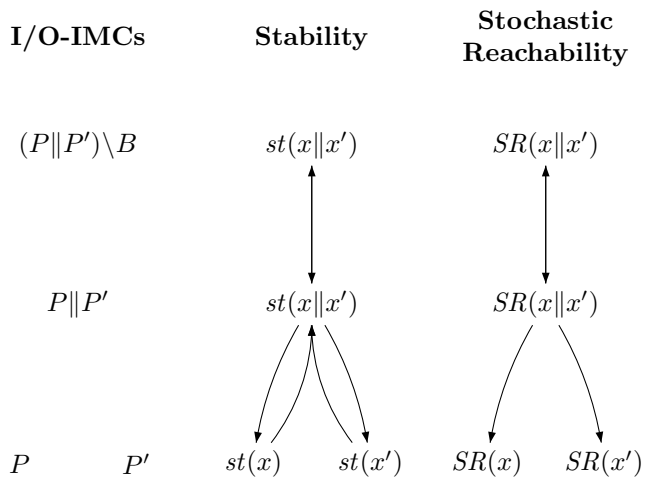


Figure 5.4: Modularity results for two compatible I/O-IMCs P and P'. Arrows indicate implications, the two arrows from $st(x)$ and $st(x')$ to $st(x\|x')$ indicate that $st(x) \wedge st(x')$ implies $st(x\|x')$

5.7 Confluence and determinism

The notions of confluence and determinism can be directly lifted from IOA to I/O-IMCs since the Markovian transitions which are – in a sense – added to IOA to obtain I/O-IMCs are deterministic (although probabilistic!) by definition.

Definition 65. *An I/O-IMC P with states S is weakly confluent (respectively weakly deterministic) if for any state $x \in S$ we have that if x is stochastically reachable, then $IOA(x)$ is weakly confluent (respectively weakly deterministic).*

The same can be applied to confluence and determinism with respect to a pair of actions.

Definition 66. *An I/O-IMC P with states S and actions A is weakly confluent (respectively weakly deterministic) with respect to a pair of actions $a, b \in A$ if for any state*

$x \in S$ we have that if x is stochastically reachable, then IOA(x) is weakly confluent (respectively weakly deterministic) with respect to a, b.

We can now combine the results on confluence and determinism for IOA with our results for stochastic reachability to find that confluence and determinism behave the same way for I/O-IMCs as for IOA.

Proposition 15. *Let P be a closed, weakly deterministic I/O-IMC P and let $[P]_{\approx}$ be its weak bisimulation quotient. For any three equivalence classes $D_1, D_2, D_3 \in S/\approx$ and any two output actions a, b, we have*

$$D_1 \longrightarrow\!\!\!\!\!\rightarrow D_2 \text{ implies } D_1 = D_2, \text{ and}$$
$$D_1 \xrightarrow{a} D_2, D_1 \xrightarrow{b} D_3 \text{ implies } a = b \text{ and } D_2 = D_3.$$

Proof. This follows directly from the equivalent proposition for IOA, Proposition 10, and the fact that stochastic reachability is preserved by weak bisimulation for I/O-IMCs. □

For the weak bisimulation quotient of a weakly deterministic I/O-IMC P we then find that any state has only a single non-divergent fair reach-trace. In Chapter 6 we will see that the consequence is that, assuming no time-divergence occurs, the I/O-IMC is indeed deterministic, i.e., its semantics is a Markov chain no matter how the "non-determinism" is resolved.

The connection between weak confluence and weak determinism is as follows. Hiding a set of pairwise weakly confluent actions "preserves" weak determinism.

Proposition 16. *Given a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ and a set of output actions $B \subset A^O$, we have that the I/O-IMC $P \backslash B$ is weakly deterministic if*

1. *for any two states $x_1, x_2 \in S$ we have that $x_1 \longrightarrow\!\!\!\!\!\rightarrow x_2$ implies $x_1 \approx x_2$,*
2. *P is* weakly confluent *with respect to all pairs of actions $a, b \in B$, and*
3. *P is* weakly deterministic *with respect to all remaining pairs of actions $a, b \in A^O \setminus B$.*

Proof. This follows directly from the equivalent proposition for IOA, Proposition 11, and the fact that hiding does not affect stochastic reachability. □

For a *complete* I/O-IMC, we only need to check the first two conditions of the above proposition. We can then make use of the fact that weak confluence is compositional.

Proposition 17. *Given $n \in \mathbb{N}$ I/O-IMCs $P_1, \ldots, P_n$, which are pairwise compatible and weakly confluent; and a set of actions B we have that if $(P_1 \| \ldots \| P_n) \backslash B$ is complete, then it is weakly deterministic.*

Proof. This follows directly from the equivalent proposition for IOA, Proposition 17, and the fact that if a state $x_1 \| \ldots \| x_n$ is stochastically reachable in the composed I/O-IMC $(P_1 \| \ldots \| P_n) \backslash B$ then its constituent states $x_1, \ldots, x_n$ are stochastically reachable in the relevant component I/O-IMCs. □

5.8 Discussion

This section reviews the contributions of the chapter, and puts them in context of related work.

The chapter has introduced I/O-IMCs as a combination of IOA and Markov chains (or more precisely a combination of IOA and infinitesimal generator matrices). We have defined parallel composition by combining the notion of parallel composition for IOA and infinitesimal generator matrices. As for IOA, shared actions are synchronised whereas different actions are interleaved. Markov transitions are always interleaved, which matches the intuitive notion of composition for CTMCs discussed in Subsection 3.3.2. We have also lifted the notion of hiding from IOA to I/O-IMCs. We have then discussed equivalences that can be used to characterize I/O-IMCs before finally discussing the concepts of stochastic reachability and determinism, both of which are highly relevant to later chapters.

5.8.1 Comparison to IMCs

I/O-IMCs are inspired by the IMCs of Hermanns [23]. IMCs are similar to I/O-IMCs except that the visible actions of an IMC are not divided into input and output actions. This means that synchronisation for IMCs is symmetric rather than asymmetric as is the case for I/O-IMCs. We now discuss the differences between the two formalisms. Any I/O-IMC can be interpreted as an IMC by simply ignoring whether visible actions are input or output actions (since this distinction does not exist for IMCs). However, by ignoring whether an action is an input or output action, states that are unstable in an I/O-IMC because of an outgoing output transition may be stable in the corresponding IMC.

Several results that hold for I/O-IMCs do not hold for IMCs. Most notably, stability *is* affected by hiding for IMCs. A state may be stable in an IMC P and unstable in the IMC $P\backslash B$. As a result, stochastic reachability is not preserved by hiding for IMCs. Similarly the modularity results for the enabledness of actions (Proposition 4) and for fair traces (Theorems 11 and 12), which I/O-IMCs inherit from IOA, do not hold for IMCs. This is rooted in our decision to base the I/O-IMC formalism on a linear-time interactive formalism (IOA), while the IMC formalism is based on a branching-time process algebra. On the other hand, I/O-IMCs are strictly less expressive than IMCs. In particular, the fact that I/O-IMCs are based on asymmetric synchronisation means that these cannot model *blocking*, i.e., two components mutually blocking the execution of the other.

5.8.2 Comparison to Wu-PIOA

Although similar, I/O-IMCs are incomparable with Wu-PIOA [53]. Wu-PIOA are also formed by combining Markovian delays and interactive transitions. However, for Wu-PIOA every output transition is tied to exactly one exponential distribution, which is not the case for I/O-IMCs, where an exponential distribution can be followed by zero or more output transitions. On the other hand, Wu-PIOA allow interactive transitions to

be equipped with a discrete probability distributions, which is not possible for I/O-IMCs. We will see in Chapter 9 that for the applications we have in mind, Wu-PIOA are too restrictive, since we cannot guarantee the strict alternation between Markovian delays and interactive transitions. In fact, many of the models used in Chapter 9 consist entirely of interactive transitions. On the other hand, discrete probability distributions are not strictly necessary for our purposes, although they may be an interesting enrichment for I/O-IMCs. Finally, it should be noted that the strict interleaving of Markovian and interactive transitions completely avoids the problem of non-determinism, which means that all Wu-PIOA can be interpreted as CTMCs.

5.8.3 I/O-IMCs as a graph-based model

In this chapter we have succeeded in introducing I/O-IMCs as an orthogonal combination of CTMCs and IOA, with the important side-note that we use the maximal progress assumption to add a notion of time to IOA (in essence, we assume that interactive events occur instantaneously, but in a particular order). As expected, the notion of parallel composition arises naturally as a combination of the notions of composition for CTMCs (see Section 3.3) and parallel composition for IOA (see Section 4.5).

In contrast to both preceeding chapters, however, we have stayed on the surface. In Chapter 3 we showed that infinitesimal generator matrics can be interpreted as Markov chains and in Chapter 4 we have seen that IOA can be interpreted as sets of fair reach-traces. The underlying semantics of I/O-IMCs is the subject of Chapter 6. Crucially, we will show that this modular semantics is sound with respect to parallel composition (as is the case for IOA and CTMCs).

6 I/O-IMC behaviours

In Chapter 3 we have seen that a regular generator matrix has as its semantics a continuous-time Markov chain, which is a specific kind of *jump process*, i.e., a stochastic process that jumps from state to state at discrete time-points. Furthermore, if we assume that two CTMCs are independent, then they can be combined in parallel in a modular fashion. In Chapter 4 we have seen that IOA also have a modular and compositionally-sound semantics in the form of sets of fair reach-traces. In Chapter 5 we have introduced the *syntax* of I/O-IMCs as an orthogonal combination of the syntax of CTMCs and IOA.

In this chapter we aim to give a modular semantics to I/O-IMCs by combining the semantics of CTMCs and IOA. This is a demanding endeavour. We will first show that I/O-IMCs also describe jump processes, albeit jump processes with a twist. To accommodate the interaction inherent in I/O-IMCs we will introduce *interactive jump processes*, which are similar to standard jump processes, except that every jump is annotated with a sequence of actions which denote the interactions that occur in that jump. In a sense, each jump of the interactive jump process can be split into two parts: a Markovian jump, and an interactive jump.

Contribution. This chapter establishes modularity results for the semantics of I/O-IMCs. They are key for providing a simple, natural, and sound semantics to I/O-IMCs. In particular, this chapter gives – for the first time – a concrete semantics for open I/O-IMCs.

6.1 Interactive jump processes

We consider a countable, possibly infinite state space S and a set of actions A partitioned into input (A^I), output (A^O), and internal (A^H) actions. Let $\perp$ denote a distinguished

state not in S as also used in Section 4.3. Let $S_\perp$ denote the extended state space $S \cup \{\perp\}$ and let $\mathcal{L}^V$ denote the set of all finite sequences of visible actions, i.e.,

$$\mathcal{L}^V = (A^I \cup A^O)^*.$$

An interactive jump process is a stochastic process $\{X^{(t)} \mid t \in \mathbb{R}_{\geq 0}\}$ which has three components. We have,

$$X^{(t)} = \langle X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}} \rangle.$$

The process X is constructed this way because at any point in time an instantaneous interactive jump may occur. The stochastic process X_{pre} takes values in $S_\perp$ and the random variable $X^{(t)}_{\mathsf{pre}}$ describes the state of X *before* the interactive jump at time t. The process X_{post} is a jump process with state space $S_\perp$, but $X^{(t)}_{\mathsf{post}}$ describes the state *after* the interactive jump at time t. Finally, we have the stochastic process W which takes values in $\mathcal{L}^V$. The random variable $W^{(t)}$ gives the names of the visible events associated with the interactive jump at time t. Essentially, the stochastic process X_{post} is a normal jump process, while the stochastic processes X_{pre} and W are used to *annotate* the jumps of X_{post}. Note that, if there is no jump at time-point t, then we have $X^{(t)}_{\mathsf{pre}} = X^{(t)}_{\mathsf{post}}$ and $W^{(t)} = \epsilon$ (the empty sequence).

Definition 67. *Given a state space S and a set of actions A, a stable* interactive jump process *is a triple $X^{(t)} = \langle X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}} \rangle, t \in \mathbb{R}_{\geq 0}$ taking values in $S_\perp \times \mathcal{L}^V \times S_\perp$ such that*

- *X_{post} is a stable jump process,*
- *X_{pre} is unequal to X_{post} for at most countably many time-points, i.e., the set*

$$\{t \mid X^{(t)}_{\mathsf{pre}} \neq X^{(t)}_{\mathsf{post}}\} \tag{6.1}$$

  is countable, and

- *W is unequal to the empty sequence ϵ for at most countably many time-points, i.e., the set*

$$\{t \mid W^{(t)} \neq \epsilon\} \tag{6.2}$$

  is countable.

We will not consider unstable interactive jump processes (i.e., where the jump process X_{post} is unstable).

As for Markov chains we will use the random variables $\{J_i \mid i \in \mathbb{N}_0\}$ to denote the jump-times of an interactive jump process X. We have $J_0 = 0$ and for all $i \in \mathbb{N}$

$$J_i = \inf\{t > J_{i-1} \mid X^{(t)}_{\mathsf{pre}} \neq X^{(J_{i-1})}_{\mathsf{post}} \vee W^{(t)} \neq \epsilon \vee X^{(t)}_{\mathsf{post}} \neq X^{(t)}_{\mathsf{pre}}\}.$$

At each jump-time $J_i, i \in \mathbb{N}$ either a Markovian jump or an interactive jump or both occur. A Markovian jump occurs whenever

$$X^{(J_i)}_{\mathsf{pre}} \neq X^{(J_{i-1})}_{\mathsf{post}}.$$

We say this is a Markovian jump *from* $X_{\mathsf{post}}^{(J_{i-1})}$ *to* $X_{\mathsf{pre}}^{(J_i)}$. An interactive jump is indicated by the fact that

$$W^{(J_i)} \neq \epsilon$$

or

$$X_{\mathsf{post}}^{(J_i)} \neq X_{\mathsf{pre}}^{(J_i)}.$$

We say this is an interactive jump from $X_{\mathsf{pre}}^{(J_i)}$ to $X_{\mathsf{post}}^{(J_i)}$ with trace $W^{(J_i)}$. Note that it is possible that the stochastic process X_{post} is unchanged by a jump, i.e., $X_{\mathsf{post}}^{(J_{i-1})} = X_{\mathsf{post}}^{(J_i)}$. It is important to note that the jumps of an interactive jump process indeed occur at discrete time-points because of the requirements we place on the stochastic processes X_{pre}, W, and X_{post}.

For the sake of convenience, we will also use so-called "time-dependent" jump-times $J_i^{(t)}$ for $i \in \mathbb{N}_0$ and $t \in \mathbb{R}_{\geq 0}$. We have $J_0^{(t)} = t$ and $J_i^{(t)}$ is the i-th jump-time after time t, i.e.,

$$J_i^{(t)} = \inf\{s > J_{i-1}^{(t)} \mid X_{\mathsf{pre}}^{(s)} \neq X_{\mathsf{post}}^{(J_{i-1}^{(t)})} \vee W^{(t)} \neq \epsilon \vee X_{\mathsf{post}}^{(t)} \neq X_{\mathsf{post}}^{(J_{i-1}^{(t)})}\}.$$

In particular, we have $J_i^{(0)} = J_i$ for all $i \in \mathbb{N}_0$. For a particular trajectory ω we write $J_i(\omega)$ and $J_i^{(t)}(\omega)$ for the corresponding jump-times for ω.

The nature of a jump is mostly determined by X_{pre}. For a jump-time J_i we have that the jump is Markovian if X_{pre} has a left-discontinuity at J_i, i.e., $\lim_{t \uparrow J_i} X_{\mathsf{pre}}^{(t)} \neq X_{\mathsf{pre}}^{(J_i)}$. The jump is interactive if X_{pre} has a right-discontinuity at J_i, i.e., $\lim_{t \downarrow J_i} X_{\mathsf{pre}}^{(t)} \neq X_{\mathsf{pre}}^{(J_i)}$, *or* W is discontinuous at J_i, i.e., $W^{(J_i)} \neq \epsilon$. If X_{pre} is both left- and right-discontinuous at J_i (or left-discontinuous and $W^{(J_i)} \neq \epsilon$) then there is a combined jump at J_i. We will show some trajectories of an interactive jump process with the different kind of jumps in Example 22.

We now consider a possible way of constructing a probability space for an interactive jump process by enumerating the jumps of the process.

6.2 Probability space

In this section we will construct a probability space for a stable interactive jump process with state space S and actions A. We describe a trajectory of this stable interactive jump process by enumerating its jumps. This approach follows Freedman's first construction for the stable case [17] and is also widely used for the construction of probability measures for CTMDPs [30]. The idea is to enumerate, for each jump, the jump time as well as the resulting state of the system. Such a description is concise and can be equipped with a probability measure by using the cross-product of standard probability measures [30]. The disadvantage of this approach is that it only describes the behaviour of the system up to the time of first explosion. In particular, if the jump times converge to some finite time-point $T < \infty$, then the enumeration of jumps does not tell us anything about the behaviour of the system after time T. This is of course another incarnation of Zeno's paradox. We will follow Freedman and deal with this form of explosion in the same

way we dealt with time-divergence for IOA, by making it explicit. For any exploding trajectory where the jump times converge to $T < \infty$ we will assume, that the system occupies the distinguished state $\perp$ for any time-point greater than or equal to T. As we have done throughout this thesis, we will refer to $\perp$ as the "time-divergence state", although we now use it to represent both time *divergence* as discussed by Hermanns [23] and jump-time *convergence* as discussed above. Note that, whenever we talk about a jump to the time-divergence state $\perp$, this is always a case of time-divergence as discussed by Hermanns. For the case that the jump-times converge, we cannot identify a particular jump to $\perp$, rather the process simply occupies $\perp$ after the time of convergence T.

The trajectories of an interactive jump processes differ from the trajectories in a jump process (such as a Markov chain) in that the jumps of an interactive jump process consist of two parts: a Markovian jump and an interactive jump. Moreover, we associate a sequence of actions from A^V with each interactive jump.

Definition 68. *Given a set of states S and a set of actions A, a* finite timed path σ *is a finite sequence of states, sequences of visible actions, states, and jump-times, i.e. for some $i \in \mathbb{N}$,*

$$\sigma \in S_\perp \times \mathcal{L}^V \times S_\perp \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^i.$$

An infinite timed path σ *is an infinite sequence of states, sequences of visible actions, states and jump-times, i.e.,*

$$\sigma \in S_\perp \times \mathcal{L}^V \times S_\perp \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^\omega.$$

We require that the sequence of jump-times is strictly increasing. The length *of a timed path equals the number of time-points in the path. For a finite timed path σ and a, possibly infinite, timed-path σ' such that the last triple of σ is equal to the first triple of σ' we write $\sigma \circ \sigma'$ for the concatenation of both paths.*

Given a timed path σ we write for an index $i \in \mathbb{N}$, $\sigma_t(i)$ for the i-th jump time of σ. By convention we write $\sigma_t(0) = 0$ and $\sigma_t(n+1) = \infty$ for a timed path of length $n \in \mathbb{N}_0$. Similarly, we write $\sigma_y(i)$ for the state after the i-th jump-time of σ, $\sigma_w(i)$ for the subsequent sequence of visible actions, and $\sigma_z(i)$ for the subsequent state. In other words we have,

$$\sigma = (\sigma_y(0), \sigma_w(0), \sigma_z(0), \sigma_t(1), \sigma_y(1), \sigma_w(1), \sigma_z(1), \ldots).$$

We will now define a sigma-algebra over the set of all timed-paths for a set of states S and a set of actions A in a similar way as is done for CTMDPs [30].

Finitely many jumps. Consider the case that the interactive jump process jumps only n times for some $n \in \mathbb{N}_0$. The trajectory of the interactive jump process is then a timed-path σ of length n. This means that after the n-th jump, the interactive jump process remains in state $last(\sigma)$ for ever.

Definition 69. *Given a number of jumps $n \in \mathbb{N}_0$, the set of all timed paths with n jumps is the set*

$$Paths_{S,A}^{(n)} = S_\perp \times \mathcal{L}^V \times S_\perp \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^n.$$

The sigma-algebra $\mathcal{F}_{S,A}^{(n)}$ on $Paths_{S,A}^{(n)}$ is the cross-product

$$\mathcal{F}_{S,A}^{(n)} = \mathcal{F}_{S_\perp} \times \mathcal{F}_{\mathcal{L}^V} \times \mathcal{F}_{S_\perp} \times \bigotimes_{i=1}^{n} (\mathcal{F}_{\mathbb{R}_{\geq 0}} \times \mathcal{F}_{S_\perp} \times \mathcal{F}_{\mathcal{L}^V} \times \mathcal{F}_{S_\perp}).$$

Here, $\mathcal{F}_{S_\perp}$ and $\mathcal{F}_{\mathcal{L}^V}$ are the standard sigma-algebras for discrete sets $S_\perp$ respectively $\mathcal{L}^V$ and $\mathcal{F}_{\mathbb{R}_{\geq 0}}$ is the Borel-algebra over the positive reals.

We write $FinPaths_{S,A}$ for the set of all finite paths, i.e.,

$$FinPaths_{S,A} = \biguplus_{n=0}^{\infty} Paths_{S,A}^{(n)},$$

where $A \uplus B$ denotes the closure under complement and countable union of the union of sets A and B. We then find the σ-algebra $\mathcal{F}_{S,A}^{fin}$ over the finite paths as the countable union of the n-jump σ-algebras, i.e.,

$$\mathcal{F}_{S,A}^{fin} = \biguplus_{n=0}^{\infty} \mathcal{F}_{S,A}^{(n)}.$$

Infinitely many jumps. We now consider the case that the interactive jump process jumps infinitely often.

Definition 70. *The set of all timed paths of infinite length is the set*

$$Paths_{S,A}^{(\infty)} = S_\perp \times \mathcal{L}^V \times S_\perp \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^\omega.$$

Given a measurable set H of finite timed-paths of length $n \in \mathbb{N}_0$, i.e., $H \subset \mathcal{F}_{S,A}^{(n)}$, the cylinder-set *of H is the set of infinite paths C_H where each infinite path in C_H is prefixed by a finite path in H, i.e.,*

$$C_H = \{\sigma \in Paths_{S,A}^{(\infty)} \mid \exists \sigma' \in H, \sigma'' \in Paths_{S,A}^{(\infty)} \cdot \sigma = \sigma' \circ \sigma''\}.$$

The σ-algebra $\mathcal{F}_{S,A}^{inf}$ is the minimal σ-algebra generated by the cylinders of the measurable sets of finite paths in $\mathcal{F}_{S,A}^{fin}$.

We now find the σ-algebra $\mathcal{F}_{S,A}$ for all timed paths as the union of the σ-algebras for finite and infinite paths.

$$\mathcal{F}_{S,A} = \mathcal{F}_{S,A}^{fin} \uplus \mathcal{F}_{S,A}^{inf}.$$

Note that $\mathcal{F}_{S,A}$ also contains sets of timed paths where the jump-times are not strictly increasing. However, we assume that such sets always have measure zero.

We will now connect the timed-paths described above to trajectories of a stable interactive jump process X. A trajectory ω of X is a function from the time-domain $\mathbb{R}_{\geq 0}$ to triples in $S_\perp \times \mathcal{L}^V \times S_\perp$. The idea is to check, for any time-point t what the last jump was before t. The state of the I/O-IMC is then simply the state resulting from that last jump. In other words, we find for a timed path σ the index i such that

$$\sigma_t(i) \leq t < \sigma_t(i+1).$$

Here we assume that the path σ has strictly increasing jump-times. Now, it becomes clear that time-convergence is a problem. If, for some infinitely long timed path we have

$$\lim_{i \to \infty} \sigma_t(i) = T$$

for some finite $T \in \mathbb{R}_{\geq 0}$, then the state of the system after time T is not defined by the timed path. For such time-convergent paths (in which the sequence of jump-times converges) we fix that, for all timed-point t greater or equal than the limit T we have that the interactive jump process occupies the distinguished state $\perp$. The same approach is used by Freedman in his first construction for the stable case [17]. With this convention we are ready to connect timed paths to trajectories of a stable interactive jump process. Recall that for a finite timed path of length n we assume $\sigma_t(n+1) = \infty$.

Definition 71. *A timed-path $\sigma \in Paths_{S,A}$ (either finite or infinite), with strictly increasing jump times fully describes a trajectory $\omega : \mathbb{R}_{\geq 0} \to S_\perp \times \mathcal{L}^V \times S_\perp$ of a stable interactive jump process X with state space S and actions A. For any time-point $t \in \mathbb{R}_{\geq 0}$, we have*

$$\omega^{(t)} = (y, w, z),$$

where

$$y = \begin{cases} \sigma_y(i), \text{ if } \sigma_t(i) = t, \\ \sigma_z(i), \text{ if } \sigma_t(i) < t < \sigma_t(i+1), \\ \perp, \text{ if } \lim_{i \to \infty} \sigma_t(i) \leq t, \end{cases}$$

$$w = \begin{cases} \sigma_w(i), \text{ if } \sigma_t(i) = t, \\ \epsilon, \text{ otherwise}, \end{cases}$$

and,

$$z = \begin{cases} \sigma_z(i), \text{ if } \sigma_t(i) \leq t < \sigma_t(i+1), \\ \perp, \text{ if } \lim_{i \to \infty} \sigma_t(i) \leq t. \end{cases}$$

It is important to note that the sample paths of X_{post} are right-continuous (as we would expect for a stable jump process), but the sample paths of X_{pre} and W may be discontinuous at countably many time-points. This matches exactly the requirements of Definition 67.

Example 22. *We now consider what a trajectory for the I/O-IMC from Example 17 (the repairable component) may look like. Imagine the following course of events: at first, the component is in good working order. Then, after five hours of running time the component breaks down, causing a "fail" event. After another two hours, a repairman manages to repair the component which is accompanied by a "repair" event. Immediately afterwards a "recover" event signifies that the component is up and running again. This scenario is described by the (partial) timed-path,*

$$\begin{aligned}
&\sigma_t(0) = 0, \sigma_y(0) = \mathbf{up}, \sigma_w(0) = \epsilon, \sigma_z(0) = \mathbf{up} \\
&\sigma_t(1) = 5, \sigma_y(1) = \mathbf{failing}, \sigma_w(1) = (\mathit{fail}), \sigma_z(1) = \mathbf{down} \\
&\sigma_t(2) = 7, \sigma_y(2) = \mathbf{down}, \sigma_w(2) = (\mathit{repair}, \mathit{recover}), \sigma_z(2) = \mathbf{up} \\
&\quad \ldots
\end{aligned}$$

Figure 6.1 shows the trajectory that describes the above scenario. Let's look at the first jump in detail. For a small time-interval $h > 0$ we have that before the jump, at time $5 - h$,

$$X_{\mathsf{pre}}^{(5-h)} = X_{\mathsf{post}}^{(5-h)} = \mathbf{up}, W^{(5-h)} = \epsilon.$$

*At time $t = 5$ our jump occurs. First we have the Markovian jump from **up** to **failing**, signified by*

$$X_{\mathsf{pre}}^{(5)} = \mathbf{failing},$$

*and then the interactive jump to **down** with a "fail" event, signified by*

$$X_{\mathsf{post}}^{(5)} = \mathbf{down}, W^{(5)} = \mathit{fail}.$$

*Afterwards we stay in state **down** for some time. We find*

$$X_{\mathsf{pre}}^{(5+h)} = X_{\mathsf{post}}^{(5+h)} = \mathbf{down}, W^{(5+h)} = \epsilon.$$

Measurability of important events. We now show that several important events of a stable interactive jump process X are indeed measurable in the probability space that we have just constructed. First, we show that the probability of observing a particular interactive jump can be measured. Secondly, we will see that also the timing of such a jump can be measured for any time-interval in the Borel-algebra on positive reals. Finally, we will use these events to compute the probability that X_{post} occupies a particular state at a particular time-point. Note that we will leave the choice of an actual probability measure for later. For now, it is enough to know that, even though I/O-IMCs are non-deterministic, an interactive jump process has a single probability measure. We will see in Section 6.3 that the non-determinism of an I/O-IMC will be represented by the fact that one I/O-IMC may have many different interactive jump processes as its semantics (similar to the way the semantics of an IOA is represented by a choice between different reach-traces).

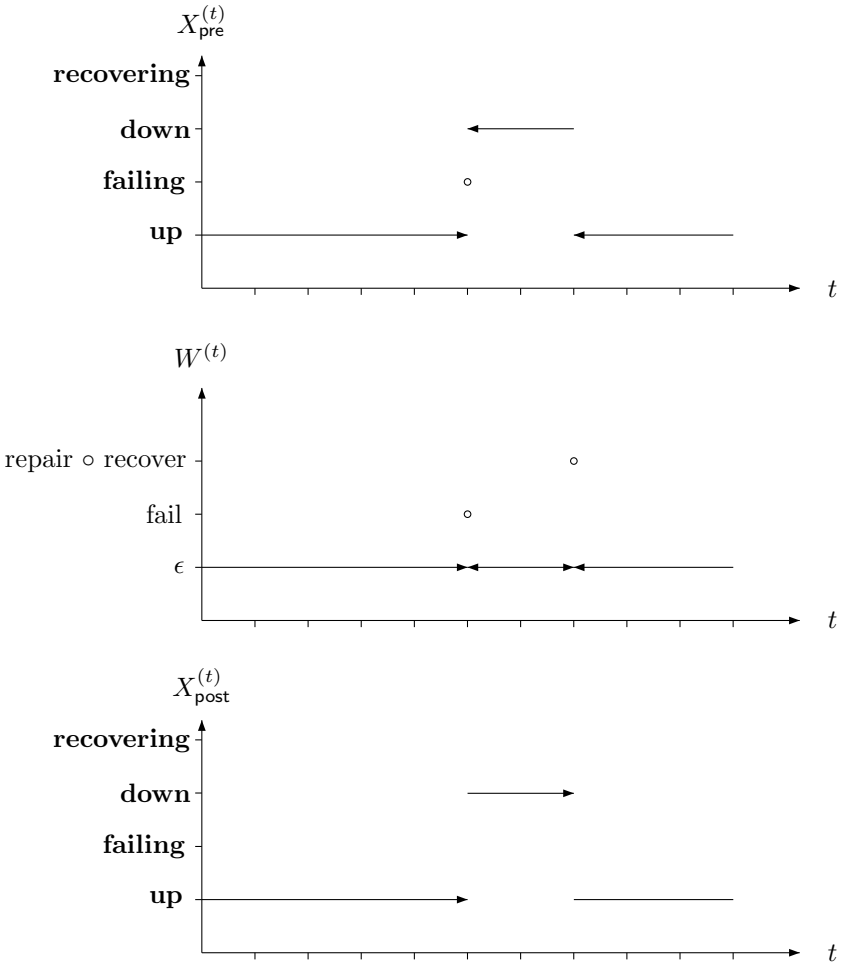


Figure 6.1: Example of a trajectory for the I/O-IMC of a repairable component.

Definition 72. *Given a state space S, a set of actions A, and a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ such that $\Omega = Paths_{S,A}$, let X be a stable interactive jump process with state space S and actions A. We will use the notation $\Pr(\ldots)$ for the probability of a certain event of X. E.g., we write*

$$\Pr(X^{(0)} = (y, w, z)) \equiv \mathcal{P}(\{\sigma \in Paths_{S,A} \mid \sigma_y(0) = y, \sigma_w(0) = w, \sigma_z(0) = z\})$$

for the probability that $X^{(0)}_{\mathsf{pre}}$ equals y, $W^{(0)}$ equals w, and $X^{(0)}_{\mathsf{post}}$ equals z for a pair of states $y, z \in S_\perp$ and a sequence $w \in \mathcal{L}^V$. Of course, we will only use this notation for events that are measurable in the probability space.

We will use a slight modification of the cylinder-set construction introduced in Definition 70. Given a measurable subset of finite paths $H \in \mathcal{F}^{fin}_{S,A}$ we consider the set C'_H of all *finite or infinite* paths, that have a prefix in H, i.e.,

$$\begin{aligned} C'_H = &\{\sigma \in Paths^{(\infty)}_{S,A} \mid \exists \sigma' \in H, \sigma'' \in Paths^{(\infty)}_{S,A} \cdot \sigma = \sigma' \circ \sigma''\} \\ &\cup \{\sigma \in FinPaths_{S,A} \mid \exists \sigma' \in H, \sigma'' \in FinPaths_{S,A} \cdot \sigma = \sigma' \circ \sigma''\}. \end{aligned}$$

It is easy to see that all such extended cylinder sets are in $\mathcal{F}_{S,A}$.

Proposition 18. *Given a stable interactive jump process X with state space S, actions A, and a probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ on the timed-paths of X, where $\mathcal{P}$ is an arbitrary probability function on $\mathcal{F}_{S,A}$, the following events are measurable.*

1. *For any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $w_i \in \mathcal{L}^V$, the set of paths where the i-th interactive jump starts in x_i, ends in y_i and has sequence w_i,*

   $$\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\}$$

   is measurable.

2. *For any time-points $t, s \in \mathbb{R}_{\geq 0}$ we have, that the set of paths where the first jump after time t occurs before time $t + s$,*

   $$\{\omega \mid J^{(t)}_1(\omega) \leq t + s\},$$

   is measurable.

3. *For any time-point $t \in \mathbb{R}_{\geq 0}$ and any state $x \in S_\perp$ we have, that the set of paths where the stochastic process X_{post} occupies state x at time t,*

   $$\{\omega \mid X^{(t)}_{\mathsf{post}} = x\},$$

   is measurable.

The proof of Proposition 18 can be found in Appendix A.1.1.

Of course, any countable conjunctions or disjunctions of the above events are also measurable. Note that there are possibly other ways of defining a probability measure for interactive jump processes. The results in the following subsections do not rely on our particular choice of probability measure, but it is important that the probabilities in Proposition 18 are indeed measurable.

6.3 I/O-IMC behaviour

In the previous section we have constructed a σ-algebra for the trajectories of a stable interactive jump process. In this section we will show how the initial distribution, interactive transition relation, and Markovian transition relation of an I/O-IMC restrict the probability function over this σ-algebra. Recall that although I/O-IMCs are non-deterministic, a single interactive jump process is in fact deterministic. The non-determinism of an I/O-IMC is illustrated by the fact that many different interactive jump process may represent the semantics of one I/O-IMC. The choice between these different possible interactive jump processes then represents the non-determinism of an I/O-IMC in the same way that the non-determinism of an IOA is represented by the choice between different reach-trace semantics.

In essence, we will given conditions on an interactive jump process such that it is a *behaviour* of an I/O-IMC P. The Markovian jumps of such a behaviour must follow the Markovian transitions of P (in the same way as the infinitesimal jump probabilities of a Markov chain must obey its infinitesimal generator) and the interactive jumps must follow the fair reach-traces of P, which are determined by the interactive transition relation (as for IOA). For a state $x \in S$ we use the short-hand notation $\mathit{FairRT}(x)$ to denote the set of fair reach-traces of the IOA rooted at x, i.e.,

$$\mathit{FairRT}(x) = \mathit{FairRT}(\mathit{IOA}(x)).$$

By definition we have $\mathit{FairRT}(\bot) = \{(\epsilon, \bot)\}$. Note that the conditions in the following definition do *not* completely determine the probability measure of an interactive jump process. In fact, they generally cannot since I/O-IMCs are non-deterministic and many different interactive jump processes will generally satisfy these conditions for any one I/O-IMC.

Definition 73. *Given an I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$, a stable interactive jump process $\{X^{(t)} = \langle X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}} \rangle \mid t \in \mathbb{R}_{\geq 0}\}$ taking values on $S_\bot \times \mathcal{L}^V \times S_\bot$ is a* behaviour *of P if and only if it satisfies the following conditions.*

1. *The distribution of X_{pre} at time 0 is given by α. For all states $x \in S$ we have*

$$\Pr(X^{(0)}_{\mathsf{pre}} = x) = \alpha_x. \tag{6.3}$$

2. *Only interactive jumps that correspond to fair reach-traces have strictly positive probability. Given a jump-index $i \in \mathbb{N}_0$, states $y, z \in S_\bot$, and a sequence $w \in \mathcal{L}^V$, such that $\Pr(X^{(J_i)}_{\mathsf{pre}} = y) > 0$, we have*

$$\Pr(X^{(J_i)}_{\mathsf{post}} = z, W^{(J_i)} = w \mid X^{(J_i)}_{\mathsf{pre}} = y) > 0 \\ \text{implies } (w, z) \in \mathit{FairRT}(y). \tag{6.4}$$

3. *The infinitesimal Markovian jump probabilities follow the Markovian transition relation. For two distinct states $x, y \in S_\bot$, a time-point $t \in \mathbb{R}_{\geq 0}$, such that*

$\Pr(X^{(t)}_{\mathsf{post}} = x) > 0$, *and a time-interval* $h > 0$ *we have*

$$\Pr(J_1^{(t)} \leq t + h, X^{(J_1^{(t)})}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x) = q_{xy}h + o(h). \tag{6.5}$$

Furthermore, we have that the probability of two jumps occurring is $o(h)$*, i.e.,*

$$\Pr(J_2^{(t)} \leq t + h \mid X^{(t)}_{\mathsf{post}} = x) = o(h). \tag{6.6}$$

4. *The probability that a Markovian jump occurs in time-interval* $[t, t+h]$ *is "independent up to* $o(h)$*" of the behaviour of* X *before time* t *and the (possible) interactive jump at* t*. For time-points* $t, t+h \in \mathbb{R}_{\geq 0}$ *and indices* $i_1, \ldots, i_n \in \mathbb{N}_0$*, a sequence of time-points* $0 < s_1, t_1, \ldots, s_n, t_n \leq t$*, states* $y_1, z_1, \ldots, y_n, z_n, x, y \in S_\perp$*, and action sequences* $w_1, \ldots, w_n \in \mathcal{L}^V$*, we have*

$$\begin{aligned}\Pr(J_1^{(t)} \leq t + h, X^{(J_1^{(t)})}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x,\\ J_{i_1} \in (s_1, t_1], X^{(J_{i_1})} = (y_1, w_1, z_1), \ldots,\\ J_{i_n} \in (s_n, t_n], X^{(J_{i_n})} = (y_n, w_n, z_n)) =\\ \Pr(J_1^{(t)} \leq t + h, X^{(J_1^{(t)})}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x) + o(h).\end{aligned} \tag{6.7}$$

We further require that (6.7) *also holds for stopping times of* X*. Recall that a random variable* T *is a stopping time if its value depends only on the trajectory of* X *up to* T*. Crucially, the jump times of* X *are stopping times.*

We denote the set of all behaviours of P *as* TrP*.*

Example 23. *Consider again the I/O-IMC from Example 17. We will construct an interactive jump process* $X = (X_{\mathsf{pre}}, W, X_{\mathsf{post}})$ *which is a behaviour of this I/O-IMC. The state space of* X *is given by the states of the I/O-IMC:*

$$S = \{\textbf{\textit{up}}, \textbf{\textit{failing}}, \textbf{\textit{down}}, \textbf{\textit{recovering}}\}$$

and the visual language is given by the visible actions of the I/O-IMC:

$$\mathcal{L}^V = \{repair, fail, recover\}.$$

The initial distribution of X *follows that of the I/O-IMC:*

$$\Pr(X^{(0)}_{\mathsf{pre}} = x) = \begin{cases} 1, & \text{if } x = \textbf{\textit{up}}, \\ 0, & \text{otherwise.} \end{cases}$$

We now choose the following transition probabilities for X:

$$\Pr(X^{(J_i)}_{\mathsf{post}} = \textbf{\textit{down}}, W^{(J_i)} = \langle \mathit{fail} \rangle \mid X^{(J_i)}_{\mathsf{pre}} = \textbf{\textit{failing}}) = 1$$
$$\Pr(X^{(J_i)}_{\mathsf{post}} = \textbf{\textit{up}}, W^{(J_i)} = \langle \mathit{repair}, \mathit{recover} \rangle \mid X^{(J_i)}_{\mathsf{pre}} = \textbf{\textit{down}}) = 1$$
$$\Pr(J_1^{(t)} \leq t+h, X^{(J_1^{(t)})}_{\mathsf{pre}} = \textbf{\textit{failing}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) = \lambda h + o(h)$$
$$\Pr(J_1^{(t)} \leq t+h, X^{(J_1^{(t)})}_{\mathsf{pre}} \neq \textbf{\textit{failing}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) = o(h)$$
$$\Pr(J_1^{(t)} \leq t+h, X^{(J_1^{(t)})}_{\mathsf{pre}} = \textbf{\textit{down}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{down}}) = \mu h + o(h)$$
$$\Pr(J_1^{(t)} \leq t+h, X^{(J_1^{(t)})}_{\mathsf{pre}} \neq \textbf{\textit{down}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{down}}) = +o(h),$$

for all $i \in \mathbb{N}$, $t, h \in \mathbb{R}$, and $h > 0$. Furthermore, X satisfies both condition 6.6 and condition 6.7 (i.e., the probability of two jumps occurring is $o(h)$ and Markovian jumps are memoryless). Lastly, we decide that the first two (interactive jump) probabilities are also memoryless (i.e., they are independent of events that occur before J_i).

It's easy to show that X satisfies the conditions of Definition 73 and is thus a behaviour of the I/O-IMC from Example 17. Let's have a closer look at the jump process X_{post}. In particular, let's investigate its infinitesimal jump probabilities:

$$\Pr(X^{(t+h)}_{\mathsf{post}} = \textbf{\textit{down}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}})$$

Since $X^{(t+h)}_{\mathsf{post}}$ is different than $X^{(t)}_{\mathsf{post}}$ there must have been a jump between t and $t+h$. We now make use of the fact that the probability that 2 jumps happens in that time-period is $o(h)$ to find that the above equals

$$\sum_{w \in \mathcal{L}^V} \sum_{y \in S} \Pr(X^{(J_1)}_{\mathsf{post}} = \textbf{\textit{down}}, W^{(J_1)} = w, J_1^{(t)}, X^{(J_1)}_{\mathsf{pre}} = y \leq t+h \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) + o(h).$$

The term $o(h)$ represents the possibility of two jumps happening in the time-period $(t, t+h]$. We can rewrite the above as

$$\sum_{w \in \mathcal{L}^V} \sum_{y \in S} \Pr(X^{(J_1)}_{\mathsf{post}} = \textbf{\textit{down}}, W^{(J_1)} = w \mid J_1^{(t)} \leq t+h, X^{(J_1)}_{\mathsf{pre}} = y, X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}})$$
$$\cdot \Pr(J_1^{(t)} \leq t+h, X^{(J_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) + o(h).$$

Applying the fact that we choose our interactive jump probabilities to be memoryless we find that this equals

$$\sum_{w \in \mathcal{L}^V} \sum_{y \in S} \Pr(X^{(J_1)}_{\mathsf{post}} = \textbf{\textit{down}}, W^{(J_1)} = w \mid X^{(J_1)}_{\mathsf{pre}} = y)$$
$$\cdot \Pr(J_1^{(t)} \leq t+h, X^{(J_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) + o(h).$$

If we now fill in the actual probabilities that we've chosen, we will see that the product inside our sum is zero (or $o(h)$) in all cases except when $w = \mathit{fail}$ and $y = \textbf{\textit{failing}}$. We

then find

$$\begin{aligned}&\Pr(X^{(J_1)}_{\mathsf{post}} = \textbf{\textit{down}}, W^{(J_1)} = \langle \mathit{fail} \rangle \mid X^{(J_1)}_{\mathsf{pre}} = \textbf{\textit{failing}})\\ &\quad \cdot \Pr(J_1^{(t)} \leq t+h, X^{(J_1)}_{\mathsf{pre}} = \textbf{\textit{failing}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{up}}) + o(h)\\ &= \lambda h + o(h).\end{aligned}$$

Similarly we find

$$\Pr(X^{(t+h)}_{\mathsf{post}} = \textbf{\textit{up}} \mid X^{(t)}_{\mathsf{post}} = \textbf{\textit{down}}) = \mu h + o(h).$$

*In fact, we will see in Section 9.3 that the X_{post} component of our interactive jump process is a continuous-time Markov chain which jumps between states **up** and **down** after exponential delays with rates λ and μ respectively. It is important to note here that the jump process X_{post} is not always a continuous-time Markov chain. In particular, there are behaviours of I/O-IMCs for which X_{post} is not memoryless.*

We now derive several more useful properties for a behaviour X of an I/O-IMC P as in Definition 73. From (6.3) it follows that

$$\Pr(X^{(0)}_{\mathsf{pre}} = \bot) = 0. \tag{6.8}$$

From (6.5) we can derive that for any time-point $t \in \mathbb{R}_{\geq 0}$, any time-length $h > 0$, and any state $x \in S_\bot$, we have

$$\Pr(J_1^{(t)} > t + h \vee X^{(J_1^{(t)})}_{\mathsf{pre}} = x \mid X^{(t)}_{\mathsf{post}} = x) = 1 - q_x h + o(h). \tag{6.9}$$

Note that the event $X^{(t)}_{\mathsf{post}} = X^{(J_1^{(t)})}_{\mathsf{pre}}$ denotes that the first jump after time t is non-Markovian. Then, (6.9) states that the probability that no *Markovian* jump occurs in h time-units is $1 - q_x h + o(h)$.

Finally, it follows from (6.7) that for time-points $t, t+h \in \mathbb{R}_{\geq 0}$ and indices $i_1, \ldots, i_n \in \mathbb{N}_0$, a sequence of time-points $0 < s_1, t_1, \ldots, s_n, t_n \leq t$, states $y_1, z_1, \ldots, y_n, z_n, x \in S_\bot$, and action sequences $w_1, \ldots, w_n \in \mathcal{L}^V$, we have

$$\begin{aligned}&\Pr(J_1^{(t)} > t + h \vee X^{(J_1^{(t)})}_{\mathsf{pre}} = x \mid X^{(t)}_{\mathsf{post}} = x,\\ &\qquad J_{i_1} \in (s_1, t_1], X^{(J_{i_1})} = (y_1, w_1, z_1), \ldots,\\ &\qquad J_{i_n} \in (s_n, t_n], X^{(J_{i_n})} = (y_n, w_n, z_n)) =\\ &\Pr(J_1^{(t)} > t + h \vee X^{(J_1^{(t)})}_{\mathsf{pre}} = x \mid X^{(t)}_{\mathsf{post}} = x) + o(h).\end{aligned} \tag{6.10}$$

Again we have that (6.10) also holds for stopping times of X.

Time-divergence. Since the pair $\langle \epsilon, \perp \rangle$ is a fair reach-trace of every IOA, we have that any behaviour of an I/O-IMC can move to the time-divergence state $\perp$ at any time. As for IOA, we can make the distinction between external time-divergence and local time-divergence. For any interactive jump from a state $y \in S$ to $\perp$ with trace w, we say that it is locally time-divergent if we can interactively reach a divergent state z in $IOA(y)$ via a trace w. Otherwise, the jump is externally time-divergent. We say a behaviour of an I/O-IMC is *non-divergent* if it makes an externally time-divergent jump with probability zero.

Definition 74. *A behaviour* $X^{(t)} = (X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}})$ *of* P *is* non-divergent *if for all jump-indices* $i \in \mathbb{N}_0$ *and all states* $x \in S$ *we have that* $\Pr(X^{(J_i)}_{\mathsf{post}} = \perp \mid X^{(J_i)}_{\mathsf{pre}} = x) > 0$ *implies that there exists a divergent state* $y \in S$ *such that* y *is reachable from* x *through a sequence of interactive transitions. In other words, whenever there is an interactive jump from a state* x *to* $\perp$ *we have that* x *can interactively reach some divergent state.*

Stability of X_{post}. We consider a behaviour $X^{(t)} = (X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}})$ of the I/O-IMC P and we focus our attention on the stochastic process X_{post}, which denotes the state of the I/O-IMC at every time-point. First, we have that this stochastic process occupies a stable state (which could be $\perp$) with probability one for any time-point smaller than the explosion time J_∞.

Proposition 19. *For any time-point* $t \in \mathbb{R}_{\geq 0}$ *we have*

$$\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\perp\} \mid J_\infty > t) = 1.$$

The proof of Proposition 19 can be found in Appendix A.1.2.

Of course, if we assume that X_{post} occupies state $\perp$ after J_∞, as we did in our construction of the probability space for interactive jump processes in, then we can drop the condition to find

$$\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\perp\}) = 1,$$

for any $t \in \mathbb{R}_{\geq 0}$.

6.4 Schedulers

As we did for Markov chains, we would like to derive the finite-jump probabilities for an I/O-IMC behaviour. However, it is clear that it is not enough to know the Markovian jump probabilities

$$\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x), x, y \in S_\perp, x \neq y, t \in \mathbb{R}_{\geq 0}, h \in \mathbb{R}_{>0}$$

which are defined – up to $o(h)$ – by (6.5), but we must also know the *interactive jump probabilities*,

$$\Pr(X^{(J_i)}_{\mathsf{post}} = z, W^{(J_i)} = w \mid X^{(J_i)}_{\mathsf{pre}} = y), x, y \in S_\perp, i \in \mathbb{N}_0,$$

which are *restricted* by (6.4), and the *external jump probabilities*

$$\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid X_{\mathsf{post}}^{(t)} = x), x \in S_\perp, t, t+h \in \mathbb{R}_{\geq 0},$$

which are not restricted at all. Furthermore, we have that the Markovian jump probabilities enjoy the Markov property – again up to $o(h)$ – because of requirement (6.7), i.e., these probabilities are independent of the past of the behaviour. Unfortunately, we have no reason to assume that the interactive jump probabilities and external jump probabilities are independent of the past. To make it easier to discuss these probabilities which may depend on the past we will now introduce the *history process* of a behaviour, which records the states and transitions the interactive jump process has visited up until a particular time-point.

6.4.1 History process

Consider a behaviour X of an I/O-IMC P with probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ as defined in Section 6.2. We now derive the stochastic process $\{Z^{(t)}, t \in \mathbb{R}_{\geq 0}\}$ which records all of the jumps of X up to time t, i.e.,

$$Z^{(t)} = \begin{cases} X^{(J_0)} \circ (J_1) \circ X^{(J_1)} \circ \ldots (J_n) \circ X^{(J_n)}, & \text{if } J_n \leq t < J_{n+1}, \\ \perp, & \text{if } J_\infty \leq t. \end{cases}$$

Recall that if X makes only n jumps then J_{n+1} equals infinity by definition. For $t = 0$ we find $Z^{(0)} = X^{(0)}$. The stochastic process Z then takes values in $FinPaths_{S,A} \cup \{\perp\}$ and we are then dealing with a stochastic process with a continuous state space. Since Z has a continuous state space we can only consider the probability that Z occupies a measurable set of states at a certain time. We will use the sigma-algebra generated by extending $\mathcal{F}^{fin}$ with the set $\{\perp\}$ to measure sets of states of Z. For instance, the probability

$$\Pr(Z^{(t)} \in \{x_0\} \times \{w_0\} \times \{y_0\} \times (3,5] \times \{x_1\} \times \{\epsilon\} \times \{x_1\})$$

is the probability that X starts in state x_0, immediately jumps to state y_0 via trace w_0, then stays in state y_0 between 3 and 5 time-units before jumping to state x_1 and staying there until time t. It is clear that this event is indeed measurable.

Before we derive the finite jump probabilities of the stochastic process Z, we first introduce some notation. Consider a time-point $t \in \mathbb{R}_{\geq 0}$, a finite timed-path of length $n \in \mathbb{N}$

$$\sigma = (x_0, w_0, y_0, t_1, x_1, w_1, y_1, \ldots, t_n, x_n, w_n, y_n)$$

such that $t_n < t$, and a sequence of time-intervals $h_1, \ldots, h_n > 0$. We are interested in the event that the jump-times of X occur "near" the time-points $t_1, \ldots, t_n$, i.e., the event

$$Z^{(t)} \in d\sigma,$$

where

$$d\sigma = \{(x_0, w_0, y_0, s_1, x_1, w_1, y_1, \ldots) \mid s_1 \in (t_1, t_1 + h_1], \ldots, s_n \in (t_n, t_n + h_n)\}.$$

Obviously this event is measurable for any choice of the intervals $h_1, \ldots, h_n$. Now, let H be any measurable event that restricts only the jumps and jump-times after the n-th jump of X. E.g., we could have

$$H = \{\sigma' \mid \sigma'_t(n+1) \in [12.4, 14.6]\}$$

describing the fact that the $n+1$-jump of X occurs between times 12.4 and 14.6. We now make the following important assumption that the limit

$$\lim_{h_1 \to 0} \cdots \lim_{h_n \to 0} \Pr(Z^{(u)} \in H \mid Z^{(t)} \in d\sigma)$$

exists for all $u > t$. By abuse of notation we will denote this limit as

$$\Pr(Z^{(u)} \in H \mid Z^{(t)} = \sigma). \tag{6.11}$$

A similar assumption is made by Doob when considering Markov chains with continuous state spaces [16]. The conditional probability in (6.11) can be interpreted as follows given that we know the transition function of X: consider a stochastic process that starts at time t and fix its history up to time t to be exactly the timed path σ. Then allow this stochastic process to evolve according to the probability function of X. The conditional probability in (6.11) is then exactly the probability that this new stochastic process behaves according to the set of paths H.

We now define the following function: $p : \mathbb{R}_{\geq 0} \times \mathit{Paths}_{S,A} \times \mathbb{R}_{\geq 0} \times \mathcal{F}_{S,A} \to [0,1]$ where

$$p(s, \sigma, u, H) = \Pr(Z^{(u)} \in H \mid Z^{(s)} = \sigma).$$

Note that for the function p we allow arbitrary sets of timed paths H, but any set of path in H that does not conform to the prefix σ will obviously have conditional probability zero. We make the further assumptions that

- $p(s, \cdot, u, H)$ is a Borel-measurable function for fixed s, u, H, and
- $p(s, \sigma, u, \cdot)$ is a probability measure on $\mathcal{F}_{S,A}$ for fixed s, σ, u.

For any time-points $s < t < u$, timed-path σ, and measurable set of timed-paths H we have

$$\begin{aligned}
&p(s, \sigma, u, H) \\
&= \Pr(H^{(u)} \in H \mid H^{(s)} = \sigma) \\
&= \int_{\mathit{Paths}_{S,A}} \Pr(H^{(u)} \in H, H^{(t)} \in d\sigma' \mid H^{(s)} = \sigma) \\
&= \int_{\mathit{Paths}_{S,A}} \Pr(H^{(u)} \in H \mid H^{(t)} = \sigma', H^{(s)} = \sigma) \Pr(H^{(t)} \in d\sigma' \mid H^{(s)} = \sigma),
\end{aligned}$$

where $\Pr(H^{(u)} \in H \mid H^{(t)} = \sigma', H^{(s)} = \sigma)$ is the obvious extension of (6.11) and we take the Lebesgue integral. That is, we select a value $c \in [0,1]$, find the measurable set of

timed-paths $d\sigma'$ such that $\Pr(H^{(u)} \in H, \mid H^{(t)} = \sigma', H^{(s)} = \sigma) = c$ for all $\sigma' \in d\sigma'$. The timed path σ' is then simply an arbitrarily chosen representative of the set $d\sigma'$. Now, if the path σ is *not* a prefix of σ' then the probability $\Pr(H^{(t)} \in d\sigma' \mid H^{(s)} = \sigma)$ must be zero. If, on the other hand, the path σ *is* a prefix of σ' then we have that $H^{(t)} = \sigma'$ implies $H^{(s)} = \sigma$. We the find that the above can be simplified to

$$\begin{aligned} &= \int_{Paths_{S,A}} \Pr(H^{(u)} \in H \mid H^{(t)} = \sigma') \Pr(H^{(t)} \in d\sigma' \mid H^{(s)} = \sigma) \\ &= \int_{Paths_{S,A}} p(t, \sigma', u, H) p(s, \sigma, t, d\sigma'). \end{aligned}$$

In short we have shown that

$$p(s, \sigma, u, H) = \int_{Paths_{S,A}} p(t, \sigma', u, H) p(s, \sigma, t, d\sigma'),$$

which is the *Kolmogorov equation* for Markov processes with continuous state spaces. We have now shown that the function p is in fact a Markov transition function as defined by Doob [16]. As a consequence we have that Z is a Markov process with a continuous state space and we will now attempt to derive finite-jump probabilities for Z as we have done for Markov chains with a countable state space.

6.4.2 Schedulers

In order to find the finite-jump probabilities of Z we must fix the interactive jump probabilities and external jump probabilities for the behaviour X. Since these probabilities may not enjoy the Markov property they may depend on the history of X. We now introduce *schedulers* which assign, for each possible history of X, interactive jump probabilities and external jump probabilities.

An *interactive jump scheduler* simply determines which reach-trace is chosen after traversing a path $\sigma \in FinPaths_{S,A}$ and then making a Markovian jump to state y (or an external jump while occupying state y) at a time-point t.

Definition 75. *Given an I/O-IMC P, an* interactive jump scheduler *is a function $f : \{\epsilon\} \cup FinPaths_{S,A} \times \mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp \to [0, 1]$ such that*

1. *$f(\cdot, \cdot, \cdot, w, y)$ is a Borel-measurable function for fixed $w \in \mathcal{L}^V$ and $y \in S_\perp$,*
2. *$f(\sigma, t, x, \cdot, \cdot)$ is a probability function on $\mathcal{L}^V \times S_\perp$ for fixed $\sigma \in \{\epsilon\} \cup FinPaths_{S,A}$, $t \in \mathbb{R}_{\geq 0}$, and $x \in S_\perp$, and*
3. *For any $\sigma \in \{\epsilon\} \cup FinPaths_{S,A}$, $t \in \mathbb{R}_{\geq 0}$, $x, y \in S_\perp$, and $w \in \mathcal{L}^V$ we have $f(\sigma, t, x, w, y) > 0$ implies $(w, y) \in FairRT(x)$.*

To emphasize that we will usually fix the first three arguments of the function f and use it as a probability function on $\mathcal{L}^V \times S_\perp$. We use the notation

$$\gamma^{(t)}_{\sigma,x}(w, y) \equiv f(\sigma, t, x, w, y),$$

for $\sigma \in \{\epsilon\} \cup FinPaths_{S,A}$, $t \in \mathbb{R}_{\geq 0}$, $x, y \in S_\perp$, and $w \in \mathcal{L}^V$.

The probability function $\gamma^{(t)}_{\sigma,x}$ describes the probability of performing different interactive jumps at time t under the condition that the interactive jump process followed path σ and made a Markovian jump to state x at time t (or an external jump when already in state x). In this sense, the interactive jump scheduler place a very similar role to the schedulers used for CTMDPs as we will see in Section 7.4. Every behaviour of an I/O-IMC P is associated with an interactive jump scheduler.

Definition 76. *If we have for a behaviour X of P and the corresponding history process Z that, for any finite timed path $\sigma \in FinPaths_{S,A}$, time-point $t \in \mathbb{R}_{\geq 0}$, states $x, y \in S_\perp$, and sequence $w \in \mathcal{L}^V$,*

$$\Pr(X^{(J_{n+1})} = (x, w, y) \mid Z^{(J_n)} = \sigma, J_{n+1} = t, X^{(J_{n+1})}_{\mathsf{pre}} = x) = \gamma^{(t)}_{\sigma,x}(w, y), \tag{6.12}$$

where this conditional probability is defined in the same way as (6.11), and

$$\Pr(X^{(J_0)} = (x, w, y) \mid X^{(J_0)}_{\mathsf{pre}} = x) = \gamma^{(0)}_{\epsilon,x}(w, y), \tag{6.13}$$

then we say that γ is the interactive jump scheduler of behaviour X.

Note, that for a path σ and time-point t, the probability function $\gamma^{(t)}_{\sigma,last(\sigma)}$ determines the interactive jump probabilities for the case that an external jump occurs at time t. The probability function $\gamma^{(0)}_{\epsilon,x}$ determines the interactive jump probabilities for the case that the behaviour starts in state x. Of course the interactive jump scheduler of a behaviour is restricted by the interactive transition relation of the associated I/O-IMC, due to requirements (6.4) and (6.12).

For the external jump probabilities we will make the following crucial assumption, that the probability of an external jump occurring in a small time-interval $(t, t+h]$ is proportionate to h (as is the case for Markovian jumps). The *external jump scheduler* then gives us the "rate" at which an external jump occurs after a particular path and at a particular time.

Definition 77. *Given an I/O-IMC P, an* external jump scheduler *is a Borel-measurable function $f : FinPaths_{S,A} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. We will use the notation*

$$\eta^{(t)}_\sigma \equiv f(\sigma, t).$$

Additionally, if for a behaviour X of P and the corresponding history process Z, we have that, for any finite timed path $\sigma \in FinPaths_{S,A}$ and time-points $t < t + h \in \mathbb{R}_{\geq 0}$,

$$Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = last(\sigma) \mid Z^{(t)} = \sigma) = \eta^{(t)}_\sigma h + o(h) \tag{6.14}$$

then we say that η is the external jump scheduler of X.

Note that a behaviour X of I/O-IMC P need not have an external jump scheduler, as Definition 73 does not require the external jump probability to be proportionate to the time-interval. However, for *non-divergent* behaviours of *closed* I/O-IMCs we have that they always have an external jump scheduler, namely one that assigns rate zero to every timed path.

Example 24. *Consider again the simple 4-state I/O-IMC from Example 17 and its behaviour X which we introduced in Example 23. We will now show that the behaviour X indeed has both an interactive jump scheduler as well as an external jump scheduler. For the interactive jump scheduler we have*

$$\gamma^{(t)}_{\sigma,x}(w,y) = \begin{cases} 1, & \text{if } x = \textbf{\textit{failing}}, w = \langle \mathit{fail} \rangle, y = \textbf{\textit{d}}, \\ 1, & \text{if } x = \textbf{\textit{down}}, w = \langle \mathit{repair}, \mathit{recover} \rangle, y = \textbf{\textit{UP}}, \\ 0, & \text{otherwise.} \end{cases}$$

For the external jump scheduler we have

$$\eta^{(t)}_{\sigma} = \begin{cases} \mu, & \text{if } \mathit{last}(\sigma) = \textbf{\textit{down}}, \\ 0, & \text{otherwise.} \end{cases}$$

We can see that both schedulers are memoryless, in that they do not consider the history of X when resolving the non-deterministic choices.

6.4.3 Finite-jump probabilities

We now consider a behaviour X of I/O-IMC P, with interactive jump scheduler γ and external jump scheduler η and the associated history process Z. We will show that the finite-jump probabilities of X (or rather of the history process Z) are completely determined by the initial distribution α of P, the infinitesimal generator matrix Q of P, the interactive jump scheduler γ, which is restricted by the interactive transition relation of P, and the external jump scheduler η, which – in a sense – is provided by the environment of P. As for Markov chains, our first step is to determine the distribution of the residence time. However, instead of computing the residence time distribution per state, we will compute the residence time distribution of X given that X has followed a particular timed path.

Lemma 16. *Given a jump-index $n \in \mathbb{N}_0$, a state $x \in S_\perp$, a finite timed path $\sigma \in \mathit{Paths}^{(n)}_{S,A}$, such that $\mathit{last}(\sigma) = x$ and $\Pr(Z^{(J_n)} = d\sigma) > 0$, and any time-point $t \in \mathbb{R}_{\geq 0}$, we have*

$$\Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) = \begin{cases} e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta^{(s)}_{\sigma}) ds}, & \text{if } \sigma_t(n) < t \\ 1, & \text{otherwise.} \end{cases} \tag{6.15}$$

Recall that $\sigma_t(n)$ is the n-th jump-time of σ.

The proof of Lemma 16 can be found in Appendix A.1.3. We can see that the residence time for an I/O-IMC behaviour is exponentially distributed just as it is for CTMCs. However, in our case the rate of the residence distribution is equal to the sum of the exit-rate q_x and the external jump rate (which may depend on the time of the jump), because X can make a jump both because of a Markovian jump and because of an external jump.

Now we are ready to compute the finite-jump probabilities of a behaviour, given that we know its interactive jump scheduler and its external jump scheduler.

Theorem 35. *Given a behaviour X (with history process Z) of I/O-IMC P with interactive jump scheduler γ and external jump scheduler η, we find for states $x, y \in S_\perp$ and a sequence of actions $w \in \mathcal{L}^V$, that*

$$\Pr(Z^{(J_0)} = (x, w, y)) = \alpha_x \gamma^{(0)}_{\epsilon,x}(w, y) \tag{6.16}$$

and for a measurable set of timed paths of length $n \in \mathbb{N}$, $H_n \in Paths^{(n)}_{S,A}$, states $y, z \in S_\perp$, and a sequence of actions $w \in \mathcal{L}^V$, we find that

$$\begin{aligned}
&\Pr(Z^{(J_{n+1})} \in H_n \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\
&\quad = \int_{t_1}^{t_2} \left(\int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds} q_{x,y} \gamma^{(t)}_{\sigma,y}(w, z) \right. \\
&\qquad \left. + \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^{t} (q_y + \eta_\sigma^{(s)}) ds} \eta_\sigma^{(t)} \gamma^{(t)}_{\sigma,y}(w, z) \right) dt.
\end{aligned} \tag{6.17}$$

Recall that $\sigma_z(n)$ is the last state of σ, since it has length n.

The proof of Theorem 35 can be found in Appendix A.1.4. Equation (6.17) is somewhat similar to Equation 3.28 which we derived for CTMCs in Chapter 3. The first term represents the probability of following a path in H_n and then making a Markovian jump ending up in z:

$$\int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \underbrace{\Pr(Z^{(J_n)} \in d\sigma)}_{\substack{\text{traverse path } \sigma \\ \text{ending in } x}} \underbrace{e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds}}_{\substack{\text{stay in } x \\ \text{until } t}} \underbrace{q_{x,y}}_{\substack{\text{Markovian jump} \\ \text{from } x \text{ to } y}} \underbrace{\gamma^{(t)}_{\sigma,y}(w, z)}_{\substack{\text{interactive jump} \\ \text{to } z \text{ with trace } w}} dt.$$

The second term represent the probability of following a path in H_n and then making an *external* jump to z:

$$\int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} \underbrace{\Pr(Z^{(J_n)} \in d\sigma)}_{\substack{\text{traverse path } \sigma \\ \text{ending in } y}} \underbrace{e^{-\int_{\sigma_t(n)}^{t} (q_y + \eta_\sigma^{(s)}) ds}}_{\substack{\text{stay in } y \\ \text{until } t}} \underbrace{\eta_\sigma^{(t)}}_{\substack{\text{External jump} \\ \text{at time } t}} \underbrace{\gamma^{(t)}_{\sigma,y}(w, z)}_{\substack{\text{interactive jump} \\ \text{to } z \text{ with trace } w}} dt.$$

Any measurable set of timed paths of length $n+1$, that is not of the form described by the left-hand side of Equation 6.17 can be described as the countable disjoint union of a number of measurable sets of paths of this form. Theorem 35 then gives us the means to recursively compute all measurable finite-jump probabilities of history process Z. However, Equation (6.17) describes the probability of observing a set of paths at the time of their last jump, not at any arbitrary time-point. Fortunately, we can derive from Equation (6.17) a recursive equation for the history process to occupy a set of paths at an arbitrary time-point.

Theorem 36. *Given a measurable set H_{n-1} of paths of length $n-1$, let $t_1 < t_2$ be two time-points, let $y, z \in S_\perp$ be two states, and let $w \in \mathcal{L}^V$ be a sequence of visible actions. For the measurable set of paths*

$$H_n = H_{n-1} \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}.$$

we find

$$\Pr(Z^{(t)} \in H_n) = \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ x = \sigma_z(n)}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_x + \eta_\sigma^{(s)}) ds}. \tag{6.18}$$

The proof of Theorem 36 can be found in Appendix A.1.5. Equation 6.18 states that the probability that the history process occupies a set of paths H_n at time t is the probability that the history process occupies the set at its n-th jump-time multiplied by the probability of not jumping until time t.

Without proof we note that from (6.17) and (6.18) it follows that

$$\begin{aligned}
&\Pr(Z^{(t)} \in H_n) \\
&\quad = \int_{t_1}^{t_2} \left(\int_{\substack{\sigma \in H_{n-1} \\ \sigma_z(n) = x \neq y}} \Pr(Z^{(s)} \in d\sigma) q_{x,y} \gamma_{\sigma,y}^{(s)}(w, z) e^{-\int_s^t (q_z + \eta_{\sigma'}^{(u)}) du} \right. \\
&\quad \left. + \int_{\substack{\sigma \in H_n \\ \sigma_z(n) = y}} \Pr(Z^{(s)} \in d\sigma) \eta_\sigma^{(s)} \gamma_{\sigma,y}^{(s)}(w, z) e^{-\int_s^t (q_z + \eta_{\sigma'}^{(u)}) du} \right) ds
\end{aligned} \tag{6.19}$$

where we write σ' for the timed path $\sigma \circ (y, w, z)$.

In this subsection we have shown that the finite-jump probabilities of the behaviour X are completely determined by the initial distribution α, the infinitesimal generator matrix Q, its interactive jump scheduler γ, and its external jump scheduler η. In the following subsection, we will attempt the reverse. Given initial distribution, infinitesimal generator matrix, interactive jump scheduler, and external jump scheduler, can we construct a behaviour X of P?

6.4.4 From scheduler to behaviour

We will now show that, given an interactive jump scheduler γ and an external jump scheduler η for I/O-IMC P, we can construct a probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ for an interactive jump process with state space S and actions A. Recall that we defined the set of timed paths $Paths_{S,A}$ and the sigma-algebra $\mathcal{F}_{S,A}$ in Section 6.2. It remains to construct the probability function $\mathcal{P}$ which assigns a probability to each measurable set of timed paths. The basis of this probability function will be the finite jump probabilities (6.16) and (6.17). Let $f_i : FinPaths_{S,A} \to [0,1], i \in \mathbb{N}_0$ be the family of additive functions induced by

$$f_0(\{(x, w, y)\}) = \alpha_x \gamma_{\epsilon,x}^{(0)}(w, y), \tag{6.20}$$

for $x, y \in S_\perp$ and $w \in \mathcal{L}^V$, and

$$\begin{aligned} & f_{n+1}(H_n \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\ & \quad = \int_{t_1}^{t_2} \left(\int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} f_n(d\sigma) e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds} q_{x,y} \gamma_{\sigma,y,w,z}^{(t)} \right. \\ & \quad \left. + \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} f_n(d\sigma) e^{-\int_{\sigma_t(n)}^{t} (q_y + \eta_\sigma^{(s)}) ds} \eta_\sigma^{(t)} \gamma_{\sigma,y,w,z}^{(t)} \right) dt, \end{aligned} \tag{6.21}$$

for $n \in \mathbb{N}$, $H_n \in Paths_{S,A}^{(n)}$, $t_1, t_2 \in \mathbb{R}_{\geq 0}$, $y, z \in S_\perp$, and $w \in \mathcal{L}^V$. It is important to note that $f(H_n)$ is understood to be the probability of the cylinder set of all (finite and infinite) paths starting with a path in H_n. In contrast, $\mathcal{P}(H_n)$ gives the probability of the finite paths in H_n, i.e., the paths which follow H_n and perform exactly n jumps. We now define $\mathcal{P}$ as the probability function induced by

$$\mathcal{P}(H_n) = f_n(H_n) - f_{n+1}(H_n \times \mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp), \tag{6.22}$$

for any index $n \in \mathbb{N}_0$ and any measurable set of paths $H_n \in Paths_{S,A}^{(n)}$, and, for a cylinder set $C(H_n)$ which consists of all infinite paths starting with a prefix in H_n,

$$\mathcal{P}(C(H_n)) = f_n(H_n) - \sum_{i=0}^{\infty} \mathcal{P}(H_n \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^i). \tag{6.23}$$

The equation (6.22) tells us that the probability of all paths in H_n is equal to the probability of all paths that *start* with a path in H_n minus the probability of those paths that start with a path in H_n and then proceed to perform any other jump. The equation (6.23) gives us that the probability of all infinite paths starting with a path in H_n equals the probability of *all* paths starting with a path in h_n minus the probability of a finite path starting in H_n. Recall that, following the first construction for the stable case of Freedman [17], we assign any "missing" probability to the state $\perp$. That is, for a timed-path σ with $t_\infty = \lim_{n\to\infty} \sigma_t(n) < \infty$ we have that the interactive jump process occupies the state $\perp$ after time-point t_∞. Without proof we note that the interactive jump process X, with history process Z, induced by the probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ indeed has interactive jump scheduler γ and external jump scheduler η and satisfies (6.16) and (6.17).

We will now show that this interactive jump process X with probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ as above is in fact a behaviour of P. To do this, we first prove that the Markovian jump probabilities of X follow the Markovian transitions of P.

Lemma 17. *Given distinct states $x, y \in S_\perp$, a path-length $n \in \mathbb{N}_0$, time-points $t < t + h \in \mathbb{R}_{\geq 0}$, and a measurable set of times paths H_n of length n, such that for each path σ in H_n $\sigma_t(n) < t$ and $\sigma_z(n) = x$ and $\Pr(Z^{(t)} \in H_n) > 0$, we have*

$$\Pr(J_{n+1} \leq t + h, X_{\mathsf{pre}}^{(J_n+1)} = y \mid Z^{(t)} \in H_n) = q_{x,y} h + o(h). \tag{6.24}$$

The proof of Lemma 17 can be found in Appendix A.1.6. We can now prove that the interactive jump process we have constructed is indeed a behaviour of the I/O-IMC P.

Theorem 37. *Given an I/O-IMC P, an interactive jump scheduler γ for P, and an external jump scheduler η for P, we have that the interactive jump process X with probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$, where $\mathcal{P}$ is constructed as per 6.22 and 6.23, is a behaviour of P.*

The proof of Theorem 37 can be found in Appendix A.1.7. Figure 6.2 illustrates our construction of an I/O-IMC behaviour from the I/O-IMC itself, an interactive jump scheduler γ, and an external jump scheduler η.

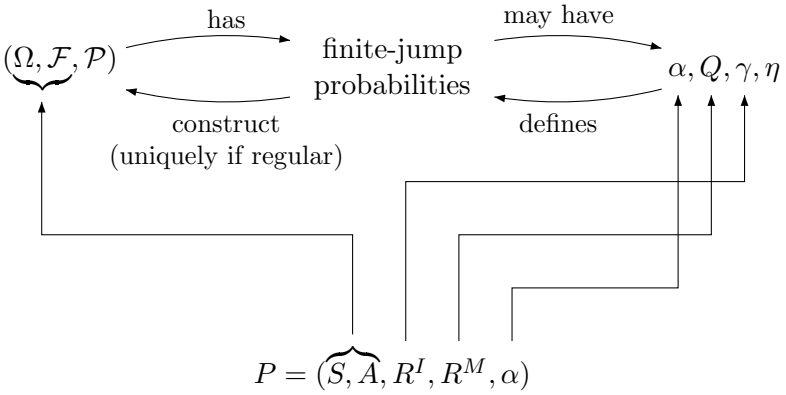


Figure 6.2: Constructing an I/O-IMC behaviour.

A few remarks are in order. First, not every behaviour can be constructed in this way. Behaviours that do not exhibit exponentially delayed external jumps will not have an external jump scheduler as in Definition 77. Furthermore, we have made the assumption that after J_∞, the time of first explosion, the behaviour always occupies the distinguished state $\perp$, but this need not be the case. In fact, as we have seen for Markov chains, the information we have (initial distribution, infinitesimal generator matrix and schedulers) does not tell us what happens after J_∞. So, as for Markov chains, if J_∞ is finite with probability greater than zero, then we may expect there to be uncountably many behaviours of P, which all have the same initial distribution, generator, and schedulers, but which all behave differently after J_∞. If, on the other hand J_∞ is infinite with probability one (i.e., the induced interactive jump process is "regular"), then we can say the initial distribution, generator, and schedulers uniquely specify the associated behaviour. In this thesis we will not consider the question whether an interactive jump process is regular and we will simply assume that all interactive jump processes we consider are regular. Furthermore we conjecture that if we restrict ourselves to the study of finite I/O-IMCs all interactive jump processes that occur will be "regular" (recall from Chapter 3 that all Markov chains on finite state spaces are regular).

So far, we have studied the properties of behaviours belonging to a single I/O-IMC. Now we turn our attention to the architectural aspects of I/O-IMCs and how they affect the associated behaviours.

6.5 Parallel composition

We now wish to show that the behaviours of I/O-IMCs are modular in the same way that executions and traces of IOA are modular. First, we define a *projection* operator for behaviours, which allows us to derive interactive jump processes for I/O-IMCs P and $\bar{P}$ from an interactive jump process of $P\|\bar{P}$. This projection operator simply combines the projection operators for states and traces and applies these to the trajectories of the interactive jump process for I/O-IMC $P\|\bar{P}$. In the remainder of this section we will consider compatible I/O-IMCs $P = \langle S, A, R^I, R^M, \alpha\rangle$, $\bar{P} = \langle \bar{S}, \bar{A}, \bar{R^I}, \bar{R^M}, \bar{\alpha}\rangle$, and their parallel composition $\tilde{P} = P\|\bar{P}$.

Definition 78. *Given an interactive jump process $\tilde{X}^{(t)} = \langle \tilde{X}^{(t)}_{\mathsf{pre}}, \tilde{W}^{(t)}, \tilde{X}^{(t)}_{\mathsf{post}}\rangle$, $t \in \mathbb{R}_{\geq 0}$ of $\tilde{P}$, the* projection *of $\tilde{X}$ onto P is the interactive jump process $\tilde{X}^{(t)} \downarrow P = \langle \tilde{X}^{(t)}_{\mathsf{pre}} \downarrow S, \tilde{W}^{(t)} \downarrow A, \tilde{X}^{(t)}_{\mathsf{post}} \downarrow S\rangle, t \in \mathbb{R}_{\geq 0}$.*

Example 25. *As an example, we look at a trajectory of a behaviour $\tilde{X}$ of the I/O-IMC $\tilde{P}$ from Figure 5.3 which represents the parallel composition of a repairable component and a repairman. The trajectory describes a failure of the component after one time-unit followed by a repair of the component after one more time-unit. Figure 6.3 shows the trajectory for $\tilde{X}$ and the accompanying trajectories for its projections X and $\bar{X}$ onto P and $\bar{P}$, respectively. In this example, the two jumps of $\tilde{X}$ are also jumps for X and $\bar{X}$. The first jump of $\tilde{X}$ is a combined jump for both $\tilde{X}$ and $\bar{X}$, but it is a purely interactive jump for X. Similarly, the second jump is a combined jump for $\tilde{X}$ and X, but a purely interactive one for $\bar{X}$.*

In general we find a strong correspondence between the jumps of a behaviour $\tilde{X}$ of $\tilde{P}$ and its projections onto P and $\bar{P}$, X respectively $\bar{X}$.

Proposition 20. *Given a stable interactive jump process $\tilde{X}$ of $P\|\bar{P}$ and its projections X and $\bar{X}$ onto P and $\bar{P}$, respectively, for every jump-index $i \in \mathbb{N}_0$ we have that there exists a jump-index $j \in \mathbb{N}_0$ such that*

$$\tilde{J}_i = J_j \text{ or } \tilde{J}_i = \bar{J}_j.$$

Moreover, the reverse also holds. For every jump-index $j \in \mathbb{N}_0$ there exist jump-indices $i, i' \in \mathbb{N}_0$ such that

$$J_j = \tilde{J}_i \text{ and } \bar{J}_j = \tilde{J}_{i'}.$$

In other words, a jump occurs for $\tilde{X}$ at any time $t \in \mathbb{R}_{\geq 0}$ if and only if a jump of X, a jump of $\bar{X}$, or a jump of both occurs at time t.

Proof. Proposition 20 follows directly from the definitions of projections and jump-times. □

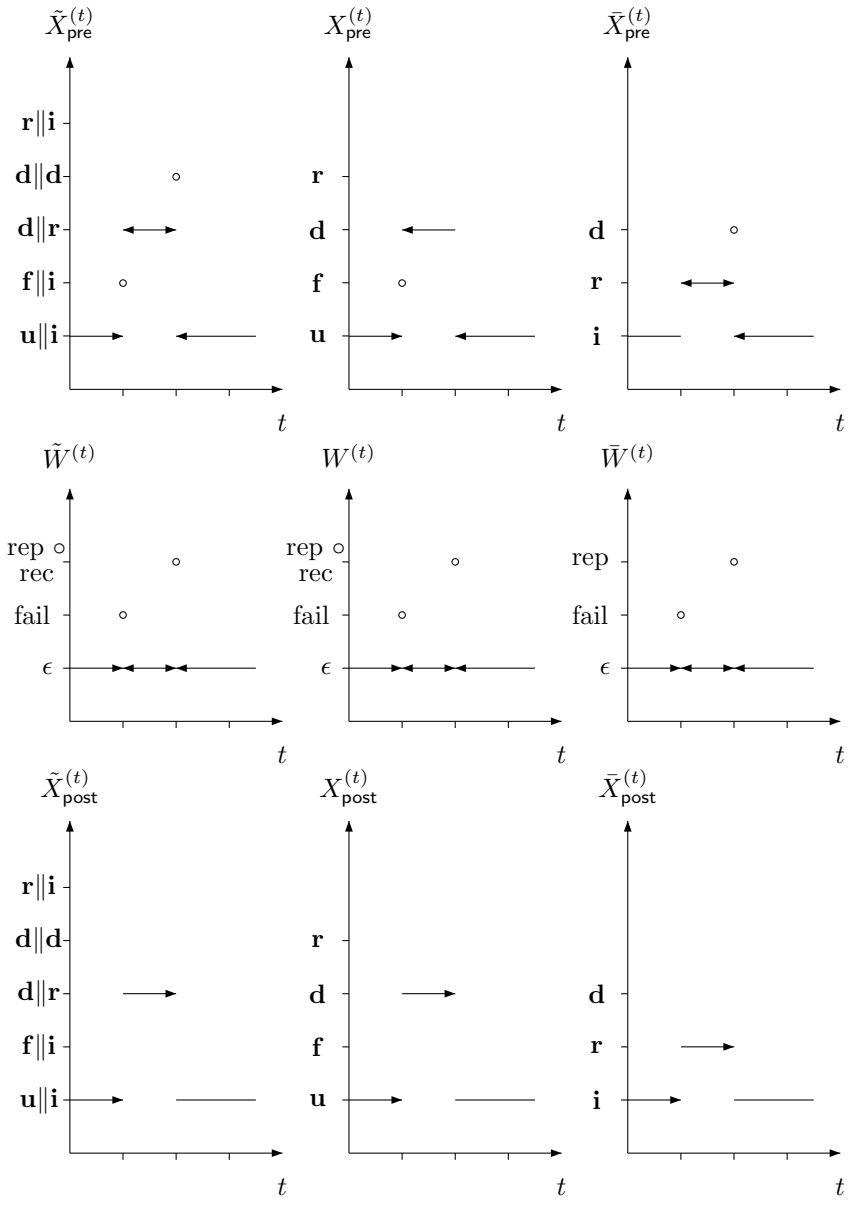


Figure 6.3: Example of a trajectory of a behaviour $\tilde{X}$ of the I/O-IMC of a repairable component composed with a repairman and its projections X and $\bar{X}$ onto the I/O-IMC of the repairable component respectively the repairman. State and action names are abbreviated.

It is important to note that both the interactive jump process $\tilde{X}$ in the above definition and both its projections $X = \tilde{X}\downarrow P$ and $\bar{X} = \tilde{X}\downarrow \bar{P}$ have the same set of trajectories and they then share the same probability space. This is in any case necessary to ensure we can consider joint events. It turns out that we can express all the important probabilities from Proposition 18 for behaviours X and $\bar{X}$ in terms of the same probabilities for behaviour $\tilde{X}$.

Proposition 21. *Given an interactive jump process $\tilde{X}$ for the I/O-IMC $\tilde{P} = P\|\bar{P}$ defined on a probability space that satisfies Proposition 18, we find that the following probabilities for the projected interactive jump process $X = \tilde{X}\downarrow P$ are measurable.*

1. *For any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $w_i \in \mathcal{L}^V$, the set of trajectories where the i-th interactive jump starts in x_i, ends in y_i and has sequence w_i,*
$$\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\},$$
is measurable.

2. *For any time-points $t, h \in \mathbb{R}_{\geq 0}$ we have, that the set of trajectories where the first jump after time t occurs before time $t + h$,*
$$\{\omega \mid J_1^{(t)}(\omega) \leq t + h\},$$
is measurable.

3. *For any time-point $t \in \mathbb{R}_{\geq 0}$ and any state $x \in S_\perp$ we have, that the set of trajectories where the stochastic process X_{post} occupies state x at time t,*
$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x\},$$
is measurable.

The proof of Proposition 21 can be found in Appendix A.1.8. As a result, it is enough to construct a probability space for $\tilde{X}$ that satisfies Proposition 18 to ensure that the important probabilities of X and $\bar{X}$ are also measurable.

We also define a notion of compatibility for behaviours. Compatible behaviours synchronize on their shared actions and are independent of each other with respect to their initial distributions and Markovian transition probabilities. Furthermore, we have that compatible behaviours should also synchronize the moment at which they experience time-divergence and we require that the probability of two jumps (for either of the behaviours) in an interval $[t, t + h]$ is $o(h)$.

Definition 79. *The two behaviours $X^{(t)} = \langle X_{\mathsf{pre}}^{(t)}, W^{(t)}, X_{\mathsf{post}}^{(t)}\rangle, t \in \mathbb{R}_{\geq 0}$ and $\bar{X}^{(t)} = \langle \bar{X}_{\mathsf{pre}}^{(t)}, \bar{W}^{(t)}, \bar{X}_{\mathsf{post}}^{(t)}\rangle, t \in \mathbb{R}_{\geq 0}$ of P respectively $\bar{P}$ are compatible if and only if,*

1. *their initial distributions are independent, i.e., for states $x \in S_\perp$, $\bar{x} \in \bar{S}_\perp$ we have*
$$\Pr(X_{\mathsf{pre}}^{(0)} = x, \bar{X}_{\mathsf{pre}}^{(0)} = \bar{x}) = \Pr(X_{\mathsf{pre}}^{(0)} = x)\Pr(\bar{X}_{\mathsf{pre}}^{(0)} = \bar{x}), \tag{6.25}$$

2. *for every time-point* $t \in \mathbb{R}_{\geq 0}$*, the words* $W^{(t)}$ *and* $\bar{W}^{(t)}$ *are synchronized with respect to the shared alphabet of* P *and* $\bar{P}$*, i.e.,*

$$W^{(t)}{\downarrow}A \cap \bar{A} = \bar{W}^{(t)}{\downarrow}A \cap \bar{A}, \tag{6.26}$$

3. *both behaviours diverge simultaneously, i.e.,*

$$X^{(t)}_{\mathsf{post}} = \bot \iff \bar{X}^{(t)}_{\mathsf{post}} = \bot, X^{(t)}_{\mathsf{pre}} = \bot \iff \bar{X}^{(t)}_{\mathsf{pre}} = \bot, \tag{6.27}$$

4. *the probability of two* distinct *jumps occurring in a time-interval* $[t, t+h]$ *with* $h > 0$ *is* $o(h)$*, i.e.,*

$$\Pr(J^{(t)}_1 \neq \bar{J}^{(t)}_1, J^{(t)}_1 \leq t+h, \bar{J}^{(t)}_1 \leq t+h) = o(h), \tag{6.28}$$

and

5. *the Markovian transition probabilities of* X *and* $\bar{X}$ *are "independent up to* $o(h)$*", i.e., for states* $x, x_1, \ldots, x_n, y, y_1, \ldots, y_n \in S_\bot$*, states* $\bar{x}, \bar{x}_1, \ldots, \bar{x}_n, \bar{y}, \bar{y}_1, \ldots, \bar{y}_n \in \bar{S}_\bot$*, action-sequences* $w_1, \ldots, w_n \in \mathcal{L}^V$*,* $\bar{w}_1, \ldots, \bar{w}_n \in \bar{\mathcal{L}}^V$*, and time-points* $t+h > t > t_1 > \ldots > t_n$*, where* $x \neq y$ *and* $\bar{x} \neq \bar{y}$*, let* H *denote the event*

$$X^{(t_1)} = (x_1, w_1, y_1), \ldots, X^{(t_n)} = (x_n, w_n, y_n)$$

and let $\bar{H}$ *denote the event*

$$\bar{X}^{(t_1)} = (\bar{x}_1, \bar{w}_1, \bar{y}_1), \ldots, \bar{X}^{(t_n)} = (\bar{x}_n, \bar{w}_n, \bar{y}_n).$$

We then require that

$$\begin{aligned}
&\Pr(J^{(t)}_1 \leq t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x, \bar{X}^{(t)}_{\mathsf{post}} = \bar{x}, H, \bar{H}) \\
&= \Pr(J^{(t)}_1 \leq t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x, H) + o(h),
\end{aligned} \tag{6.29}$$

and

$$\begin{aligned}
&\Pr(\bar{J}^{(t)}_1 \leq t+h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \mid X^{(t)}_{\mathsf{post}} = x, \bar{X}^{(t)}_{\mathsf{post}} = \bar{x}, H, \bar{H}) \\
&= \Pr(\bar{J}^{(t)}_1 \leq t+h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \mid \bar{X}^{(t)}_{\mathsf{post}} = \bar{x}, \bar{H}) + o(h),
\end{aligned} \tag{6.30}$$

and

$$\begin{aligned}
&\Pr(J^{(t)}_1 \leq t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y, \bar{J}^{(t)}_1 \leq t+h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \\
&\qquad \mid X^{(t)}_{\mathsf{post}} = x, \bar{X}^{(t)}_{\mathsf{post}} = \bar{x}, H, \bar{H}) \\
&= \Pr(J^{(t)}_1 \leq t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x, H) \\
&\qquad \cdot \Pr(\bar{J}^{(t)}_1 \leq t+h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \mid \bar{X}^{(t)}_{\mathsf{post}} = \bar{x}, \bar{H}) + o(h).
\end{aligned} \tag{6.31}$$

Given two compatible behaviours X and $\bar{X}$ as above, the fifth requirement for compatibility allows us to derive the following useful properties. In the following the variables are as in the fifth requirement of Definition 79. For the probability that X makes a Markovian jump in a time-interval $(t, t+h]$ but $\bar{X}$ does not, we find

$$\begin{aligned}
&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y, (\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x}) \\
&\qquad \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, H, \bar{H}) \\
&= \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x, H) \\
&\qquad \cdot \Pr(\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x} \mid \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, \bar{H}) + o(h).
\end{aligned} \tag{6.32}$$

We also find the reverse

$$\begin{aligned}
&\Pr((J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x), \bar{J}_1^{(t)} \leq t+h, \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{y} \\
&\qquad \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, H, \bar{H}) \\
&= \Pr(J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid X_{\mathsf{post}}^{(t)} = x, H) \\
&\qquad \cdot \Pr(\bar{J}_1^{(t)} \leq t+h, \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{y} \mid \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, \bar{H}) + o(h).
\end{aligned} \tag{6.33}$$

And finally we find a similar result for the probability that both X and $\bar{X}$ do not perform a Markovian jump:

$$\begin{aligned}
&\Pr((J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x), (\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x}) \\
&\qquad \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, H, \bar{H}) \\
&= \Pr(J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid X_{\mathsf{post}}^{(t)} = x, H) \\
&\qquad \cdot \Pr(\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x} \mid \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, \bar{H}) + o(h).
\end{aligned} \tag{6.34}$$

Again, we must ensure that the important probabilities from Proposition 18 are measurable for both compatible behaviours. The easiest way to accomplish this is to use a probability space for interactive jump processes of $\tilde{P}$, since Proposition 21 ensures us that interactive jump processes for P and $\bar{P}$ are indeed measurable in this probability space. In fact, we will now show that there is a strong connection between the behaviours of $\tilde{P}$ and pairs of compatible behaviours of P and $\bar{P}$.

6.5.1 Modularity of behaviours

First we show that a behaviour of a composed I/O-IMC can always be decomposed into compatible behaviours of its constituent I/O-IMCs. We again consider two compatible I/O-IMCs P and $\bar{P}$ and their parallel composition $\tilde{P} = P \| \bar{P}$.

Theorem 38. *Given a behaviour $\tilde{X}^{(t)}, t \in \mathbb{R}_{\geq 0}$ of $\tilde{P}$, its projections onto P and $\bar{P}$ are compatible behaviours of P and $\bar{P}$ respectively.*

The proof of Theorem 38 can be found in Appendix A.1.9. We have now shown that our semantics of I/O-IMCs is sound with respect to parallel composition: there is no behaviour of a composite I/O-IMC that cannot be projected back onto behaviours of its components. An important consequence is that parallel composition with another I/O-IMC restricts the set of possible behaviours for that I/O-IMC. If we now consider the possible transient distributions for an I/O-IMC in a parallel composition we find an interesting result.

Proposition 22. *Given a behaviour $\tilde{X}$ of $\tilde{P}$ and its projection onto P, X, we have that, for any subset of states $U \subset S$ of P*

$$\inf_{\tilde{X} \in beh(\tilde{P})} \Pr(\tilde{X}^{(t)}_{\mathsf{post}} \in \tilde{U}) \geq \inf_{X \in beh(P)} \Pr(X^{(t)}_{\mathsf{post}} \in U) \tag{6.35}$$

and

$$\sup_{\tilde{X} \in beh(\tilde{P})} \Pr(\tilde{X}^{(t)}_{\mathsf{post}} \in \tilde{U}) \leq \sup_{X \in beh(P)} \Pr(X^{(t)}_{\mathsf{post}} \in U) \tag{6.36}$$

where $\tilde{U} = \{x \| \bar{x} \mid x \in U, \bar{x} \in \bar{S}\}$.

Proof. We prove Proposition 22 by contradiction. We will consider (6.35) first. Assume then that there exists a behaviour $\tilde{X}$ of $\tilde{P}$ such that

$$\Pr(\tilde{X}^{(t)}_{\mathsf{post}} \in \tilde{U}) < \inf_{X \in beh(P)} \Pr(X^{(t)}_{\mathsf{post}} \in U).$$

It is clear from the definition of projection for behaviours that we have

$$\Pr(\tilde{X}^{(t)}_{\mathsf{post}} \in \tilde{U}) = \Pr(\tilde{X}^{(t)}_{\mathsf{post}} \downarrow P \in U).$$

But now Theorem 38 gives us that $\tilde{X}_{\mathsf{post}} \downarrow P$ is a behaviour of P, but then

$$\Pr(\tilde{X}^{(t)}_{\mathsf{post}} \downarrow P \in U) < \inf_{X \in beh(P)} \Pr(X^{(t)}_{\mathsf{post}} \in U)$$

is a contradiction. We can show a similar result for (6.36). □

Proposition 22 means that, if we can determine bounds on the transient probability distribution for some I/O-IMC P (considering all possible behaviours), then those bounds will still apply when we consider P in parallel composition with one or more other I/O-IMCs. As a result we will be able to prove properties of a complex dependable system (modelled by a parallel composition of many I/O-IMCs), by analysing only a subset of these I/O-IMCs. We will see an example of this in Section 9.3.

The reverse of Theorem 38 also holds when considering *compatible behaviours*.

Theorem 39. *Given a stable interactive jump process $\tilde{X}$ for $P \| \bar{P}$, if the projections of $\tilde{X}$ onto P and $\bar{P}$ are compatible behaviours of P and $\bar{P}$, respectively, then $\tilde{X}$ is a behaviour of $P \| \bar{P}$.*

The proof of Theorem 39 can be found in Appendix A.1.10. Note that Theorem 39 only applies to pairs of *compatible* behaviours, not arbitrary pairs of behaviours. In fact, for a composite I/O-IMC $\tilde{P} = P\|\bar{P}$, we may find a behaviour X of P such that no behaviour of $\bar{X}$ is compatible with it, which also means that there is no behaviour of $\tilde{P}$ which can be projected to yield X. The consequence is that the greater-equals sign in (6.35) cannot be replaced by equality (and the same for the lesser-equals sign in (6.36)).

Finally, we can combine Theorems 38 and 39 to characterise the behaviours of a composite I/O-IMC.

Corollary 11. *A stable interactive jump process $\tilde{X}$ of $P\|\bar{P}$ is a behaviour of $P\|\bar{P}$ if and only if its projections $\tilde{X}\downarrow P$ and $\tilde{X}\downarrow\bar{P}$ are compatible behaviours of P respectively $\bar{P}$.*

6.5.2 Modularity of schedulers

We have shown that there is a strong connection between the behaviours of compatible I/O-IMCs and behaviours of their parallel composition. We now investigate how this affects the schedulers of these behaviours. We consider two compatible I/O-IMCs P and $\bar{P}$, their parallel composition $\tilde{P} = P\|\bar{P}$, a behaviour $\tilde{X}$ of $\tilde{P}$, and its projections $X = \tilde{X}\downarrow P$ and $\bar{X} = \tilde{X}\downarrow\bar{P}$ which, as we have seen, are compatible behaviours of P respectively $\bar{P}$. We assume that all three behaviours are defined on a probability space $(Paths_{\tilde{S},\tilde{A}}, \mathcal{F}_{\tilde{S},\tilde{A}}, \mathcal{P})$ as described in Section 6.2. We first define projection for finite timed paths.

Definition 80. *The projection of a path of $\tilde{P}$ of length zero onto P is defined by*

$$(y\|\bar{y}, \tilde{w}, z\|\bar{z})\downarrow P = (y, \tilde{w}\downarrow P, z),$$

where $y, z \in S_\perp$, $\bar{y}, \bar{z} \in \bar{S}_\perp$, and $\tilde{w} \in \tilde{\mathcal{L}}^V$. For a finite timed path of length greater than zero $\tilde{\sigma} = \tilde{\sigma}' \circ (t, y\|\bar{y}, \tilde{w}, z\|\bar{z})$, with $\tilde{\sigma}' \in FinPaths_{\tilde{S},\tilde{A}}$, $t \in \mathbb{R}_{\geq 0}$, $y, z \in S_\perp$, $\bar{y}, \bar{z} \in \bar{S}_\perp$, and $\tilde{w} \in \tilde{\mathcal{L}}^V$, where $x = last(\tilde{\sigma}')$ we have

$$\tilde{\sigma}\downarrow P = \begin{cases} \tilde{\sigma}'\downarrow P, & \text{if } x = y = z, \tilde{w}\downarrow P = \epsilon \\ \tilde{\sigma}'\downarrow P \circ (t, y, \tilde{w}\downarrow P, z), & \text{otherwise.} \end{cases}$$

The projection of paths of $P\|\bar{P}$ onto $\bar{P}$ are defined similarly.

Consider a set of finite timed paths H_n of P of length n of the form

$$\{(x_0, w_0, y_0)\} \times (s_1, t_1] \times \{(x_1, w_1, y_1)\} \times \ldots \times (s_n, t_n] \times \{(x_n, w_n, y_n)\},$$

for the states $x_0, y_0, \ldots, x_n, y_n$, the sequences $w_0, \ldots, w_n \in \mathcal{L}^V$, and the time-points $s_1, t_1, \ldots, s_n, t_n \in \mathbb{R}_{\geq 0}$. Obviously, this set of paths is in $\mathcal{F}_{S,A}$. It will turn out to be important to consider which finite timed paths of $P\|\bar{P}$ project onto a path in H_n. By studying Definition 80 it is clear that for each jump of P, there may occur any number

of jumps of $\tilde{P}$ (as long as these do not affect the state of P or involve actions in the alphabet of P). We then find

$$
\begin{aligned}
&\{\tilde{\sigma} \subset \mathit{FinPaths}_{\tilde{S},\tilde{A}} \mid \tilde{\sigma}{\downarrow}P \in H_n\} \\
&= \bigcup_{i_0=0}^{\infty} \cdots \bigcup_{i_n=0}^{\infty} \\
&\{(x_0, w_0, y_0)\} \\
&\times (\mathbb{R}_{\geq 0} \times \{y_0 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\} \times \{\tilde{w} \in \tilde{\mathcal{L}}^V \mid \tilde{w}{\downarrow}P = \epsilon\} \times \{y_0 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\})^{i_0} \\
&\times (s_1, t_1] \times \{(x_1, w_1, y_1)\} \\
&\times (\mathbb{R}_{\geq 0} \times \{y_1 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\} \times \{\tilde{w} \in \tilde{\mathcal{L}}^V \mid \tilde{w}{\downarrow}P = \epsilon\} \times \{y_1 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\})^{i_1} \\
&\qquad \vdots \\
&\times (s_n, t_n] \times \{(x_n, w_n, y_n)\} \\
&\times (\mathbb{R}_{\geq 0} \times \{y_n \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\} \times \{\tilde{w} \in \tilde{\mathcal{L}}^V \mid \tilde{w}{\downarrow}P = \epsilon\} \times \{y_n \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\})^{i_n}.
\end{aligned}
$$

The natural numbers $i_0, \ldots, i_n$ describe the number of jumps of $\tilde{P}$, which do not affect P, that occur after each jump of P. The rectangle

$$(\mathbb{R}_{\geq 0} \times \{y_0 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\} \times \{\tilde{w} \in \tilde{\mathcal{L}}^V \mid \tilde{w}{\downarrow}P = \epsilon\} \times \{y_0 \| \bar{y} \mid \bar{y} \in \bar{S}_\perp\})^{i_0}$$

for instance describes i_0 jumps that do not affect P, since neither the state of P changes, nor do the sequence of actions contain any actions in the alphabet of P.

We now turn to the schedulers of X, $\bar{X}$, and $\tilde{X}$. We will assume that $\tilde{X}$ has an interactive jump scheduler $\tilde{\gamma}$ and an external jump scheduler $\tilde{\eta}$. Since the behaviours X and $\bar{X}$ are defined on the sigma-algebra $\mathcal{F}_{\tilde{S},\tilde{A}}$ we first consider their schedulers in terms of paths of $\tilde{P}$.

Theorem 40. *If $\tilde{X}$ has interactive jump scheduler $\tilde{\gamma}$, then we find for the interactive jump probabilities of behaviour X, that*

$$
\begin{aligned}
&\Pr(X_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y, W^{(\tilde{J}_{i+1})} = w \mid X_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = x, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma}) \\
&= \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}{\downarrow}P = w}} \sum_{\bar{y} \in \bar{S}_\perp} \tilde{\gamma}^{(t)}_{\tilde{\sigma}, x \| \bar{x}}(\tilde{w}, y \| \bar{y}),
\end{aligned}
\tag{6.37}
$$

for a jump-index $i \in \mathbb{N}_0$, a path $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, states $x, y \in S_\perp$, $\bar{x} \in \bar{S}_\perp$, a sequence $w \in \mathcal{L}^V$, and a time-point $t \in \mathbb{R}_{\geq 0}$. Moreover, for the function $f : \left(\{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}\right) \times \mathbb{R}_{\geq 0} \times \tilde{S}_\perp \times \mathcal{L}^V \times S_\perp \to [0,1]$ defined as

$$f(\tilde{\sigma}, t, x, w, y) \equiv \Pr(X_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y, W^{(\tilde{J}_{i+1})} = w \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = x \| \bar{x}, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma})$$

we find that

1. $f(\cdot,\cdot,\cdot,w,y)$ *is a Borel-measurable function for fixed* $w \in \mathcal{L}^V$ *and* $y \in S_\perp$,
2. $f(\tilde{\sigma},t,\tilde{x},\cdot,\cdot)$ *is a probability function on* $\mathcal{L}^V \times S_\perp$ *for fixed* $\tilde{\sigma} \in \{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, $t \in \mathbb{R}_{\geq 0}$, *and* $\tilde{x} \in \tilde{S}_\perp$, *and*
3. *For any* $\tilde{\sigma} \in \{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, $t \in \mathbb{R}_{\geq 0}$, $x \in \tilde{S}_\perp$, $y \in S_\perp$, *and* $w \in \mathcal{L}^V$ *we have* $f(\sigma,t,\tilde{x},w,y) > 0$ *implies* $(w,y) \in \mathit{FairRT}(\tilde{x}{\downarrow}P)$.

We find a similar result for the interactive jump probabilities of $\bar{X}$.

The proof of Theorem 40 can be found in Appendix A.1.11. We find a similar result for the external jump scheduler of $\tilde{X}$.

Theorem 41. *If* $\tilde{X}$ *has external jump scheduler* $\tilde{\eta}$, *then we find for the external jump probabilities of* X *that*

$$\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid \tilde{Z}^{(t)} = \tilde{\sigma}) = \left(\sum_{\substack{\bar{y} \in \bar{S}_\perp \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}_{\tilde{\sigma}}^{(t)} \right) h + o(h) \tag{6.38}$$

for states $x \in S_\perp$, $\bar{x} \in \bar{S}_\perp$, *a path* $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$ *with* $\mathit{last}(\tilde{\sigma}) = x\|\bar{x}$ *and a time-point* $t \in \mathbb{R}_{\geq 0}$. *Moreover, we have that the function* $f : \mathit{FinPaths}_{\tilde{S},\tilde{A}} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *defined by*

$$f(\tilde{\sigma},t) = \begin{cases} \sum_{\substack{\bar{y} \in \bar{S}_\perp \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}_{\tilde{\sigma}}^{(t)}, & \text{if } \mathit{last}(\tilde{\sigma}) = x\|\bar{x} \\ 0, & \text{if } \mathit{last}(\tilde{\sigma}) = \perp. \end{cases}$$

is Borel-measurable. We find a similar result for the external jump scheduler $\bar{\eta}$ *of* $\bar{X}$.

The proof of Theorem 41 can be found in Appendix A.1.12.

We can see that there is a discrepancy between the schedulers which we defined for the I/O-IMC P in Section 6.4 and the schedulers we found in Theorems 40 and 41 by projection. The former schedulers fix interactive jump probabilities and external jump probabilities for paths in $\mathit{FinPaths}_{S,A}$, but the schedulers we have just derived in the aforementioned theorems fix these probabilities for paths in $\mathit{FinPaths}_{\tilde{S},\tilde{A}}$. In principal, this is not a problem, since we could define the probability space of the behaviour of P to act on the paths induced by the composite I/O-IMC $\tilde{P}$. However, this somehow takes away from our modularity results as we then cannot consider an I/O-IMC P in isolation; we would always need to know with what other I/O-IMCs it will be composed in order to study its schedulers (for instance, to calculate lower and upper bounds for P to reach a certain state).

To overcome this problem, we will now try to derive schedulers for P which act on the paths induced by P (rather than the paths induced by $P\|\bar{P}$) from the schedulers for $P\|\bar{P}$. We will do this by determining the jump probabilities for paths in $\mathit{FinPaths}_{S,A}$ instead of $\mathit{FinPaths}_{\tilde{S},\tilde{A}}$.

Lemma 18. *Let X, $\bar{X}$, and $\tilde{X}$ be behaviours of P, $\bar{P}$, and $\tilde{P} = P \| \bar{P}$ respectively and let $\tilde{\gamma}$ and $\tilde{\eta}$ be the interactive jump respectively external jump scheduler of $\tilde{X}$. Given a path $\sigma \in FinPaths_{S,A}$, a time-point $t \in \mathbb{R}_{\geq 0}$, states $x, y \in S_\perp$, a sequence of actions $w \in \mathcal{L}^V$, and a jump-index $i \in \mathbb{N}_0$ we find for the interactive jump probabilities of X that*

$$\begin{aligned}
&\Pr(X^{(J_{i+1})}_{\mathsf{post}} = y, W^{(J_{i+1})} = w \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma) \\
&= \sum_{k=i}^{\infty} \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma} \in H_k} \sum_{\substack{\tilde{y} \in \tilde{S}_\perp \\ \tilde{y}\downarrow P = y}} \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, \tilde{y}) \\
&\quad \cdot \Pr(\tilde{J}_{k+1} = J_{i+1}, \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma). \qquad (6.39)
\end{aligned}$$

Furthermore, given a path $\sigma \in FinPaths_{S,A}$, a time-point $t \in \mathbb{R}_{\geq 0}$, and a state $x \in S_\perp$, we find for the external jump probabilities of X that

$$\begin{aligned}
&\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = x \mid Z^{(t)} = \sigma) \\
&= \left(\sum_{\bar{x} \in \bar{S}} \int_{\substack{\tilde{\sigma} \in H \\ last(\tilde{\sigma}) = x \| \bar{x}}} \left(\sum_{\substack{\bar{y} \in \bar{S} \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}^{(t)}_{\tilde{\sigma}} \right) \Pr(\tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma) \right) h + o(h). \qquad (6.40)
\end{aligned}$$

The proof of Lemma 18 can be found in Appendix A.1.13. This result seems promising. We can simply choose as our schedulers for P, the schedulers γ and η which assign interactive and external jump probabilities according to Equations (6.39) and (6.40) respectively, i.e.,

$$\begin{aligned}
\gamma^{(t)}_{\sigma,x}(w, y) = &\sum_{k=i}^{\infty} \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma} \in H_k} \sum_{\substack{\tilde{y} \in \tilde{S}_\perp \\ \tilde{y}\downarrow P = y}} \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, \tilde{y}) \\
&\cdot \Pr(\tilde{J}_{k+1} = J_{i+1}, \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma)
\end{aligned}$$

and

$$\eta^{(t)}_{\sigma} = \sum_{\bar{x} \in \bar{S}} \int_{\substack{\tilde{\sigma} \in H \\ last(\tilde{\sigma}) = x \| \bar{x}}} \left(\sum_{\substack{\bar{y} \in \bar{S} \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}^{(t)}_{\tilde{\sigma}} \right) \Pr(\tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma).$$

The question remains whether γ and η satisfy the other requirements of Definitions 75 and 77. From Theorem 40 it easily follows that $\gamma^{(t)}_{\sigma,x}$ is indeed a probability function for fixed σ, x, and t and that it assigns positive probability only to fair reach-traces of x. But, it is not clear that $\gamma^{(\cdot)}_{\cdot,\cdot}(w, y)$ for fixed w, y and $\eta^{(\cdot)}$ are Borel-measurable functions. However, we conjecture that we can always find schedulers γ' and η' for P such that the finite-jump probabilities induced by these schedulers lie arbitrarily close to the finite-jump probabilities of X (which we can compute using schedulers $\tilde{\gamma}$ and $\tilde{\eta}$ and the projection of paths of $\tilde{P}$ onto P).

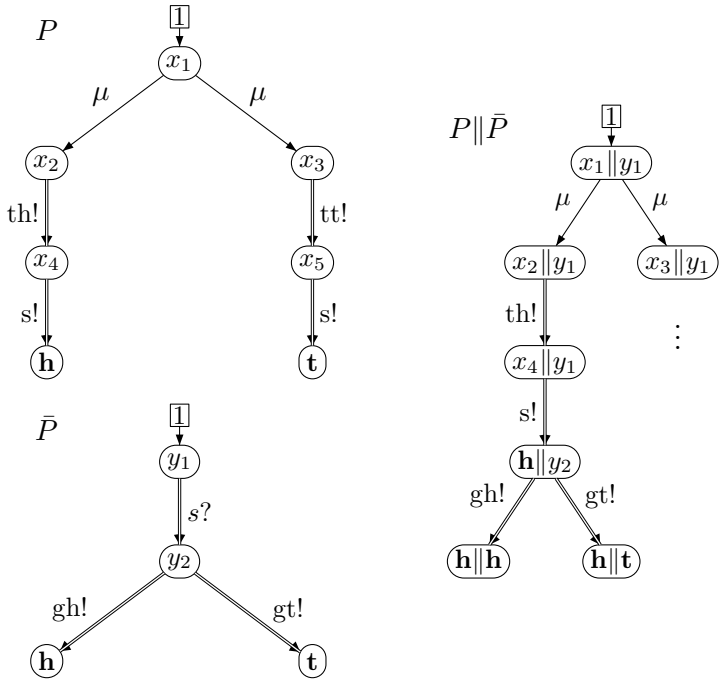


Figure 6.4: Two I/O-IMCs showing a coin-toss experiments and (part of) their parallel composition. States and actions in the parallel composition have been abbreviated.

Example 26. *Let us see the projection of schedulers in action with an example that describes a coin-tossing experiment. Figure 6.4 shows two I/O-IMCs P and $\bar{P}$ and their parallel composition $\tilde{P} = P\|\bar{P}$. The I/O-IMC P represents a person tossing a fair coin and the I/O-IMC $\bar{P}$ represents another person guessing (afterwards!) whether the outcome is heads or tails. Now, let us try to find the scheduler which maximizes the probability that we've guessed correctly, i.e., the scheduler that maximizes the probability of reaching states $\boldsymbol{h}\|\boldsymbol{h}$ or $\boldsymbol{t}\|\boldsymbol{t}$. The easiest way of doing this is to simply base our guess on the last state of the path of $\tilde{P}$: if we are in state $\boldsymbol{h}\|y_4$ pick the transition labelled gh! to state $\boldsymbol{h}$ and if we are in state $\boldsymbol{t}\|y_4$ pick the transition labelled gt! to state $\boldsymbol{t}$:*

$$\tilde{\gamma}^{(t)}_{\tilde{\sigma},\boldsymbol{h}\|y_4}(\tilde{w},\tilde{y}) = \begin{cases} 1, & \text{if } \tilde{w} = \langle gh! \rangle, \tilde{y} = \boldsymbol{h}\|\boldsymbol{h}, \\ 0, & \text{otherwise.} \end{cases}$$

$$\tilde{\gamma}^{(t)}_{\tilde{\sigma},\boldsymbol{t}\|y_4}(\tilde{w},\tilde{y}) = \begin{cases} 1, & \text{if } \tilde{w} = \langle gt! \rangle, \tilde{y} = \boldsymbol{t}\|\boldsymbol{t}, \\ 0, & \text{otherwise.} \end{cases}$$

All other values of $\tilde{\gamma}$ are chosen arbitrarily.

Now we can project this scheduler onto $\bar{P}$ using (6.39)*. In essence, we find that the probability of $\tilde{P}$ reaching state $\boldsymbol{h}\|y_4$ is exactly $1/2$. The same goes for state $\boldsymbol{h}\|y_4$. This*

means that, the projection of $\tilde{\gamma}$ will pick action gh! with probability one half:

$$\bar{\gamma}^{(t)}_{\sigma,y_4}(w,y) = \begin{cases} 1/2, & \text{if } w = \langle gh! \rangle, y = \boldsymbol{h}, \\ 1/2, & \text{if } w = \langle gt! \rangle, y = \boldsymbol{t}, \\ 0, & \text{otherwise.} \end{cases}$$

So we have seen that we can derive scheduler $\bar{\gamma}$ for $\bar{P}$ from scheduler $\tilde{\gamma}$ for $P \| \bar{P}$, preserving the stochastic behaviour of $\bar{P}$, that is, the probability to reach $\boldsymbol{h}$ in $\bar{P}$ using scheduler $\bar{\gamma}$ is the same as the probability to reach the set of states $S \times \{\boldsymbol{h}\}$ in $P \| \bar{P}$ using scheduler $\tilde{\gamma}$. However, the reverse is not *true. Selecting scheduler $\bar{\gamma}$ for $\bar{P}$ does not infer that we must select $\tilde{\gamma}$ for $P \| \bar{P}$, in fact this would be very unintuitive, since it would mean that randomly guessing the outcome of a coin (scheduler $\bar{\gamma}$) would be the same as always guessing the outcome of the fair coin correctly (scheduler $\tilde{\gamma}$). It is interesting to note that the reverse does hold: if we only observe the "guesser" and cannot see the outcome of the coin-toss we cannot tell the difference between a person who always guesses correctly (scheduler $\tilde{\gamma}$) and a person guessing randomly (scheduler $\bar{\gamma}$).*

Example 26 shows us a very important point: although we conjecture that we can always project the schedulers of a composite I/O-IMC onto its components to yield "local" schedulers (i.e., defined on the paths of the component I/O-IMC) which preserve the stochastic behaviour of the component I/O-IMC, we cannot do the reverse. Given two local schedulers for the component I/O-IMCs we cannot compose these to find a scheduler of the composite I/O-IMC. This is directly related to the fact that for IOA, we cannot compose traces of component IOA to find a trace of a composite IOA (see Section 4.5). It is important to remember that if we define the probability space of the behaviours of the component I/O-IMCs to be the same as the probability space of the composite I/O-IMC, then we *can* compose behaviours of the component I/O-IMCs to find a behaviour of the composite I/O-IMC. We conjecture that we can do the same for the schedulers of these behaviours.

Figure 6.5 gives an overview of the modularity results for I/O-IMC behaviours. We will revisit the discussion of which probability space to use for I/O-IMCs in a parallel composition in Section 6.7.3. Among other things, we will discuss the connection to the work of Giro and D'Argenio on distributed schedulers [20].

6.6 Hiding

We also extend the hiding operator to interactive jump processes.

Definition 81. *Given an I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$, a set of output actions $B \subseteq A^O$, as well as a stable interactive jump process $X^{(t)} = (X^{(t)}_{\mathsf{pre}}, W^{(t)}, X^{(t)}_{\mathsf{post}})$ with $t \in \mathbb{R}_{\geq 0}$ for P,* hiding *the actions B in X, denoted – by abuse of notation – $X \backslash B$, results in the stable interactive jump process $X^{(t)} \backslash B = (X^{(t)}_{\mathsf{pre}}, W^{(t)} \backslash B, X^{(t)}_{\mathsf{post}})$ with $t \in \mathbb{R}_{\geq 0}$, where $W^{(t)} \backslash B$ is the projection of $W^{(t)}$ onto the visible actions of $P \backslash B$, i.e.,*

$$W^{(t)} \backslash B = W^{(t)} \downarrow (A^V \setminus B).$$

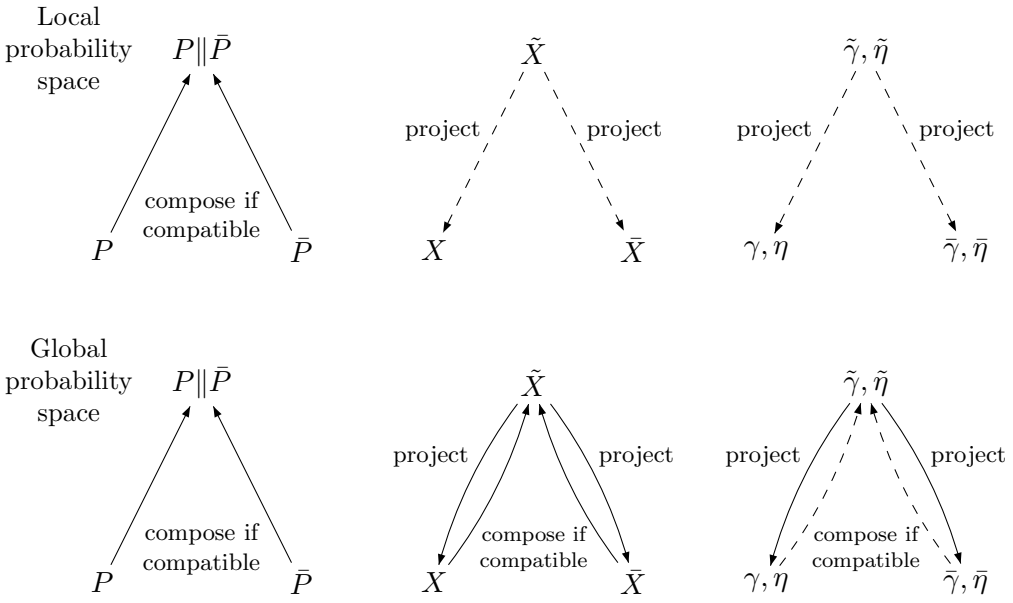


Figure 6.5: Modularity results for I/O-IMC behaviours. At the top we see results for the case that the probability spaces of component behaviours are based on the paths of the component I/O-IMCs. Below we see the results for the case that the probability spaces of component behaviours are based on the paths of the composite I/O-IMC. Dotted arrows represent conjectures.

Clearly, $X \backslash B$ is a stable interactive jump process of $P \backslash B$. We will now show that the jumps of $X \backslash B$ coincide with the jumps of X with probability one. In the following we assume an I/O-IMC P as above and a subset of its output actions B.

Proposition 23. *Given a behaviour X of P, let $\bar{X} = X \backslash B$. For any time-point $t \in \mathbb{R}_{\geq 0}$, we then have*

$$\Pr(\bar{J}_1^{(t)} \neq J_1^{(t)}) = 0.$$

The proof of Proposition 23 can be found in Appendix A.1.14.

Proposition 24. *Given an interactive jump process X for the I/O-IMC P defined on a probability space that satisfies Proposition 18, we find that the following probabilities for the abstracted interactive jump process $\bar{X} = X \backslash B$ are measurable.*

1. *For any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $\bar{w}_i \in \bar{\mathcal{L}}^V$, the set of trajectories where the i-th interactive jump starts in x_i, ends in y_i and has sequence w_i,*

$$\{\omega \mid \bar{X}^{(\bar{J}_i)}(\omega) = (x_i, \bar{w}_i, y_i)\},$$

is measurable.

2. *For any time-points $t, h \in \mathbb{R}_{\geq 0}$ we have, that the set of trajectories where the first jump after time t occurs before time $t + h$,*

$$\{\omega \mid \bar{J}_1^{(t)}(\omega) \leq t + h\},$$

is measurable.

3. *For any time-point $t \in \mathbb{R}_{\geq 0}$ and any state $x \in S_\perp$ we have, that the set of trajectories where the stochastic process $\bar{X}_{\mathsf{post}}$ occupies state x at time t,*

$$\{\omega \mid \bar{X}_{\mathsf{post}}^{(t)}(\omega) = x\},$$

is measurable.

Proof. Since, with probability one, the jumps of $\bar{X}$ correspond to the jumps of X we find for the first probability, that

$$\{\omega \mid \bar{X}^{(\bar{J}_i)}(\omega) = (x_i, \bar{w}_i, y_i)) = \cup_{\substack{w_i \in \mathcal{L}^V : \\ w_i \backslash B = \bar{w}_i}} \{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\}.$$

The second and third sets of trajectories are trivially measurable in the probability space of X. □

We now show that hiding actions in a behaviour X of I/O-IMC P is "safe", i.e., the resulting interactive jump process is a behaviour of $P \backslash B$.

Theorem 42. *Given an I/O-IMC P, a subset of its output actions B, and a behaviour X of P, we have that $X \backslash B$ is a behaviour of $P \backslash B$.*

The proof of Theorem 42 can be found in Appendix A.1.15.

Theorem 43. *Given an I/O-IMC P, a subset of its output actions B, and a behaviour $\bar{X}$ of $P \backslash B$, we have that there exists a behaviour X of P such that $\bar{X} = X \backslash B$.*

The proof of Theorem 43 can be found in Appendix A.1.16.

We can now connect the behaviours of an I/O-IMC with the behaviours of its abstraction. Other than for parallel composition, we can see that abstracting actions in an I/O-IMC removes information. Given a behaviour of an abstracted I/O-IMC $P \backslash B$ we only know that there exists a representative behaviour of P.

Corollary 12. *Given, an I/O-IMC P and a subset of its output actions B, a stable interactive jump process $\bar{X}$ is a behaviour of $P \backslash B$ if and only if there exists a behaviour X of P such that $X \backslash B = \bar{X}$.*

6.7 Discussion

In this chapter we have provided a modular semantics for I/O-IMCs by combining the semantics of IOA and CTMCs in an orthogonal way. Markovian transitions are treated as completely independent of each other and interactive transition sequences are synchronized on their shared alphabets. The two aspects (Markovian and interactive transitions) are combined by defining every jump to have two parts: a Markovian part and an interactive part. The interactive jumps are assumed to occur instantaneously (this assumption is also known as the maximal progress assumption [39]), which means that they do not interfere with the Markovian jumps, since the probability of an instantaneous Markovian jump is zero (for stable Markov chains).

6.7.1 Relationship to CTMCs

In Chapter 3, we have seen that a regular infinitesimal generator matrix induces a CTMC. In particular, the entries in the infinitesimal generator matrix describe the probability to jump from one state to another in a small time-period. Similarly, the Markovian transitions in I/O-IMCs also describe the infinitesimal jump probabilities of the underlying interactive jump process. However, an I/O-IMC does *not* induce a single interactive jump process, but a family of them. This is caused by the fact that there is uncertainty in when interactive transitions (initiated by other I/O-IMCs running in parallel) occur, or which interactive transitions are taken when multiple locally controlled actions are enabled.

It is important to note that this uncertainty does not affect what we know about Markovian transitions in stable states. Consider the I/O-IMC in Figure 6.6. We can see that in the initial state x, there is a choice between a Markovian transition to state y with rate λ and an interactive transition to state z labelled with input action a. Under the assumption that the action a is controlled by another I/O-IMC, the probability to jump from state x to state y is still given by the rate on the Markovian transition:

$$\Pr(X^{(t+h)}_{\mathsf{post}} = y \mid X^{(t)}_{\mathsf{post}} = x) = \lambda h + o(h).$$

This follows from (6.5) and the fact that in this case we cannot reach any other stable states from state y. This means that, despite the presence of interactive transitions, the Markovian transitions of I/O-IMCs still behave the same as the transitions in a CTMC and the above statement holds for every interactive jump process induced from the I/O-IMC in Figure 6.6.

6.7.2 Relationship to IOA

Consider a parallel composition of three I/O-IMCs $P_1 \| P_2 \| P_3$. Recall that the jumps of an interactive jump process induced from this composite I/O-IMC consist of two parts: the Markovian jump and the interactive jump (see Figure 6.7). Let us look at just the interactive part of one of these jumps where these three I/O-IMCs start in three states x, y, and z respectively. Now, the possible interactions that occur between

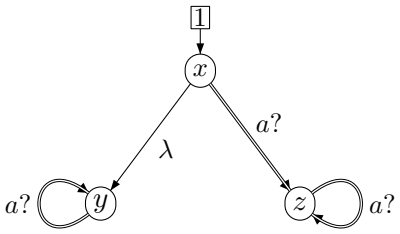


Figure 6.6: Example of an I/O-IMC.

these three I/O-IMCs are exactly the possible interactions between the IOA rooted at states x, y, and z (recall from Definition 48 that this is just the IOA constructed by picking, e.g., x as starting state and then adding all reachable states and interactive transitions). This means that the outcome of the interactive phase of a jump in an I/O-IMC is completely independent of the Markovian transitions of that I/O-IMC and inherits all the properties of IOA (or more precisely, the variant of IOA we introduced in Chapter 4), most importantly the modularity of its trace semantics. To find out what the possible interactions are for our composite I/O-IMC we can simply compose in parallel the three IOA rooted at x, y, and z respectively.

Example 27. *Figure 6.7 gives an example of how we can determine the possible interactive jumps for a composite I/O-IMC $P_1 \| P_2 \| P_3$ where P_1 just made a Markovian jump from $\tilde{x}$ to x. We first consider the IOA rooted at x, y, and z respectively and then take their parallel composition. As we can see, there is a non-deterministic choice whether to go to state $\tilde{x} \| \tilde{y} \| z$ with action sequence ab or state $\tilde{x} \| y \| \tilde{z}$ with action sequence ac.*

6.7.3 Global and local schedulers

There is one important caveat to be made regarding the way we resolve non-determinism using schedulers in this Chapter which we have touched upon in Example 26. Consider a composed I/O-IMC $\tilde{P} = P \| \bar{P}$. We have shown that any behaviour of $\tilde{P}$ can be projected onto P and $\bar{P}$ and conversely compatible behaviours of P and $\bar{P}$ can be combined to construct a behaviour of $\tilde{P}$. However, these operations are only possible if all three behaviours are defined on the same sigma-algebra, namely that of $\tilde{P}$. This causes a problem when we try to compose and decompose schedulers. The schedulers of a behaviour of P need to assigns probabilities for all paths of $\tilde{P}$. This is counter-intuitive as it means the non-determinism in P can be resolved by looking at the state and history of $\bar{P}$. Depending on what we are modelling, we might expect that the decisions made by P should depend only on the history of P itself. Furthermore, we would like to study the behaviour of P in isolation, without having to take into consideration which I/O-IMCs it is composed in parallel with.

One possible way to overcome this problem is to use *distributed schedulers* [20]. Distributed schedulers are a restricted class of schedulers that ensure that local decisions are based on local information, even in a non-local setting. This means that the schedulers of $\tilde{P}$ must resolve non-deterministic decisions that are local to P using only the

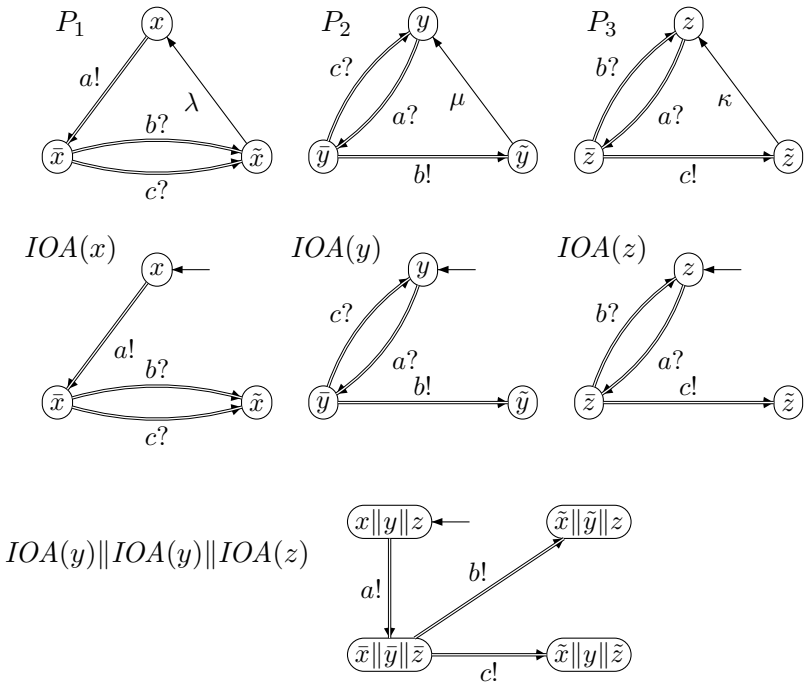


Figure 6.7: Example of an I/O-IMC and a close look at the possible interactive jumps starting in state $x\|y\|z$.

path-information of P. It can be expected that such schedulers can be projected directly onto the sigma-algebra for P. Unfortunately, even distributed schedulers use global information, namely to resolve non-determinism between components (i.e., whenever both P and $\bar{P}$ have actions enabled).

7 Closed behaviours

In the previous chapter we have given a natural, modular semantics to I/O-IMCs by combining the natural semantics of Markov chains and IOA. In this chapter we will turn our attention to the subset of *closed* I/O-IMCs, i.e., I/O-IMCs that cannot interact. We will show that closed I/O-IMCs correspond to continuous-time Markov decision processes (CTMDP) [28], which means that the modular semantics of Chapter 6 matches the monolithic translational semantics adapted from Johr [30]. The correspondence between closed I/O-IMCs and CTMDPs allow us to reuse existing analysis techniques for CTMDPs to analyse closed I/O-IMCs.

Recall that the goal of I/O-IMCs has been to give a compositional way to construct CTMCs. The reason we arrive at CTMDPs instead of CTMCs is the inherent non-determinism of IOA, which is inherited by I/O-IMCs. For I/O-IMCs that do not exhibit non-determinism we would then expect that they correspond to CTMCs. In Section 7.6 we will show that this is indeed the case: a deterministic I/O-IMC induces a single interactive jump process which is a Markov chain (to be more exact, the stochastic process X_{post} of the interactive jump process is a Markov chain).

Contribution This chapter revolves around the notion of closed behaviours (behaviours that exhibit no external jumps), which give a semantic underpinning of closed I/O-IMCs. It establishes several important claims from earlier chapters: that weak bisimulation preserves the transient distributions associated with I/O-IMCs and that a state of an I/O-IMC is *stochastically reachable* if and only if the state can be reached with non-zero probability for some resolution of the non-determinism. We further show the correspondence between closed I/O-IMCs and CTMDPs. Finally, we demonstrate that the unique behaviour of a closed deterministic I/O-IMC is indeed a CTMC. This may not come as a surprise, but in our setting this result is not obvious and not straightforward to demonstrate. This is caused by our deliberate decision to avoid a monolithic

semantic interpretation.

7.1 Basic definition

In this section we will give the basic definition of a *closed behaviour* of an I/O-IMC and we will show several simple properties of such behaviours.

A *closed behaviour* of an I/O-IMC is a behaviour that exhibits no external jumps. In essence, this means that the behaviour is not influenced by its environment. Throughout this chapter we will assume that such behaviours belong to closed I/O-IMCs. In principle, it is possible that a behaviour of an open I/O-IMC exhibits no external jumps (e.g., if it is composed in parallel with a complementary I/O-IMC that never interacts with it), but we do not discuss this possibility here. We consider a closed I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$.

Definition 82. *A behaviour X of the closed I/O-IMC P is called closed if it exhibits an external jump with probability zero. That is, for any jump-index $i \in \mathbb{N}_0$ and any state $x \in S_\perp$, such that $\Pr(X^{(J_i)}_{\mathsf{post}} = x) > 0$ we have*

$$\Pr(X^{(J_{i+1})}_{\mathsf{pre}} = x \mid X^{(J_i)}_{\mathsf{post}} = x) = 0.$$

Our first observation is that any non-divergent behaviour of a closed I/O-IMC P is a closed behaviour. This means, that the only way an external jump may happen for a behaviour of a closed I/O-IMC is through time-divergence, i.e., external jumps only occur if a behaviour running in parallel jumps to the distinguished state $\perp$.

Proposition 25. *Given a behaviour X of closed I/O-IMC P we have that if X is non-divergent then X is closed.*

The proof of Proposition 25 can be found in Appendix A.2.1. We now consider a closed behaviour X of the closed I/O-IMC P defined on the probability space described in Section 6.2. Since the probability of an external jump for X is zero, we have that its external jump scheduler is constantly zero. This means that the recursive derivation of the finite-jump probabilities can be simplified.

Theorem 44. *Given a closed behaviour X (with history process Z) of I/O-IMC P with interactive jump scheduler γ, we find for states $x, y \in S_\perp$ and a sequence of actions $w \in \mathcal{L}^V$, that*

$$\Pr(Z^{(J_0)} \in \{(x, w, y)\}) = \alpha_x \gamma^{(0)}_{\epsilon,x}(w, y) \tag{7.1}$$

and for a measurable set of timed paths of length $n \in \mathbb{N}$, $H_n \in Paths^{(n)}_{S,A}$, states $y, z \in S_\perp$, and a sequence of actions $w \in \mathcal{L}^V$, we find that

$$\begin{aligned}&\Pr(Z^{(J_{n+1})} \in H_n \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\ &\quad = \int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \Pr(Z^{(J_n)} \in d\sigma) e^{-q_x(t - t_n)} q_{x,y} \gamma^{(t)}_{\sigma,y}(w, z) dt.\end{aligned} \tag{7.2}$$

Proof. Theorem 44 follows directly from substituting $\eta_\sigma^{(t)} = 0$ into (6.17). □

Similarly we find, for a measurable set of timed paths of length $n \in \mathbb{N}$, $H_n \in Paths_{S,A}^{(n)}$, states $y, z \in S_\perp$, and a sequence of actions $w \in \mathcal{L}^V$, that

$$\begin{aligned}&\Pr(Z^{(t)} \in H_{n-1} \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\ &\quad = \int_{t_1}^{t_2} \int_{\substack{\sigma \in H_{n-1} \\ \sigma_z(n) = x \neq y}} \Pr(Z^{(s)} \in d\sigma) q_{x,y} \gamma_{\sigma,y}^{(s)}(w, z) e^{-q_z(t-s)} ds. \end{aligned} \tag{7.3}$$

This follows from substituting $\eta_\sigma^{(t)} = 0$ into (6.19).

Closed behaviours represent a non-deterministic Markovian process *without* compositionality. It is then no surprise that such behaviours are closely related to closed interactive Markov chains and continuous-time Markov decision processes, which are both non-deterministic and non-compositional Markovian models. In subsection 7.4 we will illustrate this connection by proposing translations from closed I/O-IMCs to CTMDPs, but first we will show some key results for closed I/O-IMCs and weak bisimulation.

7.2 Weak bisimulation

In this section we will prove the claim that weak bisimulation preserves the transient state-distributions induced by I/O-IMCs. We first consider a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ and a weak bisimulation relation $\mathcal{E}$ on S. As so often, we will consider the finite jump probabilities of a behaviour X of P defined by the interactive jump scheduler γ. However, this time we consider jumps between *equivalence classes* of $\mathcal{E}$, rather than between states. This approach is similar to our approach to bisimulation for Markov chains (recall Section 3.2). Let $K_i, i \in \mathbb{N}$ be the times when X jumps between equivalence classes of $\mathcal{E}$. That is,

$$K_0 = 0$$

and for all $i > 0$,

$$K_i = \inf\{t > K_{i-1} \mid X_{\mathsf{pre}}^{(t)} \not\mathcal{E} X_{\mathsf{post}}^{(J_{i-1})} \vee W^{(t)} \neq \epsilon \vee X_{\mathsf{post}}^{(t)} \not\mathcal{E} X_{\mathsf{pre}}^{(t)}\}. \tag{7.4}$$

We will assume that the number of state-jumps in between two equivalence-class-jumps is always finite with probability one. This in essence means that we assume the interactive jump process X to be "regular". This assumption can for instance be realized by restricting to finite I/O-IMCs. We have seen in our discussion of bisimulation for CTMCs (see Section 3.2), that bisimulation does not preserve transient distributions for irregular Markov chains (unless the equivalence classes are chosen appropriately) and we assume that the same holds for interactive jump processes derived from I/O-IMCs, hence our assumption that we are only dealing with "regular" interactive jump processes.

As we did for CTMCs in Chapter 3 we define the "infinitesimal generator matrix" $\bar{Q}$ over the equivalence classes of $\mathcal{E}$. For two equivalence classes $D, D' \in S/\mathcal{E}$ we have

$$\bar{q}_{D,D'} = \begin{cases} \sum_{y \in D'} q_{x,y}, & \text{if } D \neq D', \text{ for arbitrary } x \in D, \\ -\sum_{y \notin D} q_{x,y}, & \text{if } D = D', \text{ for arbitrary } x \in D. \end{cases}$$

For convenience we write $\bar{q}_D = -\bar{q}_{D,D}$.

We begin by considering the distribution of the residence time of a behaviour X of P with respect to the equivalence classes of $\mathcal{E}$. That is, we are interested in the distribution of K_{i+1}, i.e., we want to know for a *stable* equivalence class D and some time-point $t \in \mathbb{R}_{\geq 0}$

$$\Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D).$$

Note that for an unstable equivalence class D' the probability $\Pr(X^{(K_i)}_{\mathsf{post}} \in D)$ is zero. Also, since stability is preserved by weak bisimulation, all states in D are stable.

Theorem 45. *Given a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ and a weak bisimulation relation $\mathcal{E}$ on S, we find for any equivalence class $D \in S/\mathcal{E}$, any time-point $t \in \mathbb{R}_{\geq 0}$, and any jump-index $i \in \mathbb{N}$, that*

$$\Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D) = 1 - e^{-\bar{q}_D t}.$$

The proof of Theorem 45 can be found in Appendix A.2.2. We can now consider the finite jump probabilities of our I/O-IMC with respect to jumps between equivalence classes. The following theorem follows the same structure as Theorem 44, but considers paths over equivalence classes, i.e., set of paths from $Paths^{(n)}_{S/\mathcal{E},A}$.

Theorem 46. *Given a closed behaviour X (with history process Z) of I/O-IMC P with interactive jump scheduler γ, we find for equivalence classes $D, D' \in S/\mathcal{E}$ and a sequence of actions $w \in \mathcal{L}^V$, that*

$$\Pr(Z^{(0)} \in D \times \{w\} \times D') = \sum_{x \in D} \sum_{y \in D'} \alpha_x \gamma^{(0)}_{\epsilon,x}(w, y) \tag{7.5}$$

and for a measurable set of timed paths (across equivalence classes) of length $n \in \mathbb{N}$, $H_n \in Paths^{(n)}_{S/\mathcal{E},A}$, equivalence classes $D', D'' \in S/\mathcal{E}$, and a sequence of actions $w \in \mathcal{L}^V$, we find that

$$\begin{aligned}&\Pr(Z^{(t)} \in H_n \times (t_1, t_2] \times \{D\} \times \{w\} \times \{D'\}) \\ &\quad = \int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \notin D}} \Pr(Z^{(s)} \in d\sigma) \sum_{y \in D} q_{x,y} \sum_{z \in D'} \gamma^{(s)}_{\sigma,y}(w, z) e^{-\bar{q}_{D'}(t-s)} ds.\end{aligned} \tag{7.6}$$

The proof of Theorem 46 follows from applying (7.1) and (7.3) to sets of paths over equivalence classes. It is crucial to realize that any jump between equivalence classes (as defined in Equation (7.4)) always starts with a Markovian jump to a distinct equivalence class (see also the proof of Theorem 45).

We now turn our attention to weakly bisimilar I/O-IMCs. We consider two complete, weakly bisimilar I/O-IMCs $P = (S, A, R^I, R^M, \alpha)$ and $\bar{P} = (\bar{S}, \bar{A}, \bar{R}^I, \bar{R}^M, \bar{\alpha})$ with disjoint state spaces. Let $\mathcal{E}$ be a weak bisimulation relation for the disjoint union of P and $\bar{P}$ that relates their initial distributions. That is,

$$\sum_{x \in D \cap S} \alpha_x = \sum_{\bar{x} \in D \cap \bar{S}} \bar{\alpha}_{\bar{x}},$$

for every equivalence class D in $S \cup \bar{S}/\mathcal{E}$. The interactive jump scheduler γ for P induces a closed behaviour X for P.

We will now show that the weakly bisimilar I/O-IMCs P and $\bar{P}$ have behaviours which are equivalent up to the equivalence classes of $\mathcal{E}$. For the sake of simplicity we write $q_{D,D'}$ for $\sum_{y \in D' \cap S} q_{x,y}$ for arbitrary $x \in D \cap S$ and $\bar{q}_{D,D'}$ for $\sum_{y \in D' \cap \bar{S}} \bar{q}_{x,y}$ for arbitrary $x \in D \cap \bar{S}$.

Theorem 47. *For every interactive jump scheduler γ of P there exists an interactive jump scheduler $\bar{\gamma}$ for $\bar{P}$ such that, for the induced behaviours X respectively $\bar{X}$ we have, for any time-point t, that*

$$\Pr(X^{(t)}_{\mathsf{post}} \in D \cap S) = \Pr(\bar{X}^{(t)}_{\mathsf{post}} \in D \cap \bar{S}).$$

The proof of Theorem 47 can be found in Appendix A.2.3, with the caveat that due to a problem regarding measurability of the scheduler we choose for $\bar{P}$, we can only prove that the transient probabilities of $\bar{X}_{\mathsf{post}}$ are arbitrarily close to those of X_{post} rather than equal.

7.3 Stochastic reachability

In this section we prove the claim we made in Section 5.6, that a stochastically reachable state can indeed be reached with probability greater than zero and vice versa. We will consider a closed I/O-IMC P with state space S. Recall that a state y of P is stochastically reachable if there exists a path from an initial state x to y which consists of either interactive transitions or Markovian transitions from stable states. We call such a path a *plausible path*. It will be useful to decompose such a plausible path into Markovian transitions and fair reach-traces.

Theorem 48. *The finite path σ starting in state $x \in S$ and ending in a stable state $y \in S$, is a plausible path if and only if there exists a length $n \in \mathbb{N}$, a sequence of states $x_1, \ldots, x_n \in S$, and a sequence of stable states $y_1, \ldots, y_n \in S$ such that*

1. *$x = x_1$ and $y = y_n$,*
2. *the states y_i are the stable states along σ, i.e.,*

$$\sigma \downarrow S_s = \langle y_1, \ldots, y_n \rangle,$$

3. *y_i is fairly reachable in $IOA(x_i)$ for all $1 \leq i \leq n$, and*
4. *there is a Markovian transition from each y_i to x_{i+1}, i.e., $y_i \hookrightarrow x_{i+1}$ for all $1 \leq i < n$.*

Note that we may have that x_i equals y_i for certain indices i.

Proof. Given that every finite path ending in a stable state is fair, Theorem 48 is easy to prove. □

Now we will show that there is a close correspondence between stochastic reachability and the finite-probabilities induced by the closed behaviours of P.

Theorem 49. *A stable state x in S is stochastically reachable if and only if there exists an interactive jump scheduler γ, which induces closed behaviour X, such that for all time-points $t \in \mathbb{R}_{\geq 0}$, with $t > 0$, the probability that X_{post} occupies x at time t using finitely many jumps is greater than zero. That is,*

$$SR(x) \Leftrightarrow \exists \gamma \cdot \forall t > 0 \cdot \Pr(X^{(t)}_{\mathsf{post}} = x, J_\infty > t) > 0. \tag{7.7}$$

The proof of Theorem 49 can be found in Appendix A.2.4.

7.4 Continuous-time Markov decision processes

Continuous-time Markov decision processes (CTMDPs) are state-based models which are both non-deterministic and stochastic (see e.g., [52]).

Definition 83. *A CTMDP is described by a four-tuple (S, A, R, x_0) where S is a finite set of states, A is a finite set of actions, $R \subset S \times A \times \mathbb{R}_{\geq 0} \times S$ is a transition relation, and $x_0 \in S$ is the initial state. For every pair $x \in S$ and $a \in A$ we have that either a is* enabled *in x and then we have*

$$\forall y \in S \cdot |\{\lambda \mid (x, a, \lambda, y)\}| = 1$$

or a is not enabled in x and then we have

$$\forall y \in S \cdot |\{\lambda \mid (x, a, \lambda, y)\}| = 0.$$

We denote the set of actions enabled in a state x as A_x. For states $x, y \in S$ and an action $a \in A$ enabled in x, we write

$$R(x, a, y) = \lambda$$

for $\lambda \in \mathbb{R}_{\geq 0}$ such that $(x, a, \lambda, y) \in R$. Similarly we write

$$R(x, a) = \sum_{y \in S} R(x, a, y).$$

Intuitively, $R(x, a, y)$ is the transition-rate from x to y under decision a and $R(x, a)$ is the exit-rate of x under decision a.

Definition 84. *We say a CTMDP is* locally uniform *if for each state x there exists a rate $\lambda \in \mathbb{R}_{\geq 0}$ such that*

$$\forall a \in A_x \cdot R(x, a) = \lambda.$$

We write $R(x)$ for the exit-rate of a state x of a locally uniform CTMDP.

We consider a CTMDP M with state space S, actions A, initial state x_0 and transition relation R. We are interested in the *timed paths* a CTMDP traverses. A timed path is a tuple which strings together transitions from one state, choosing a particular action and at a particular time, to another state.

Definition 85. *The set of all finite timed paths* CPaths *of* M *is given by*

$$\mathsf{CPaths}_M = \bigcup_{n=0}^{\infty} S \times (A \times \mathbb{R}_{\geq 0} \times S)^n .$$

We will leave out the subscript when it is clear from context which CTMDP is meant.

The length of a path is equal to the number of transitions in the path. That is, the path $\langle x \rangle$ with $x \in S$ has length zero and the path $\langle x, a, t, y \rangle$, with $a \in A$, $t \in \mathbb{R}_{\geq 0}$, and $y \in S$ has length one, etc. Although we do not consider them here, the results presented in this subsection also extend to infinitely long paths [52]. As an example, the path $\langle x_0, a_1, t_1, x_1, a_2, t_2, x_2 \rangle$ represents the following sequence of events. Starting in state x_0, the action a_1 is selected. The CTMDP then jumps from state x_0 to state x_1 at time t_1. Subsequently, the action a_2 is selected and the CTMDP jumps to state x_2 at t_2. Note that the times t_1 and t_2 represent *jump-times* (similar to our treatment of timed paths for I/O-IMCs) and not *residence times* as in the work of Wolovick and Johr [52]. We do this purely for the sake of simplicity; the two approaches are equivalent.

We use the following notations. For a path of length $n \in \mathbb{N}$,

$$\sigma = \langle x_0, a_1, t_1, x_1, \ldots, a_n, t_n, x_n \rangle$$

we write $\sigma_a(i) = a_i$, $\sigma_t(i) = t_i$, and $\sigma_x(i) = x_i$ for the i-th action, jump-time, and state respectively.

We want to find a probability measure over the set of all timed paths. That is, we want to be able to make statements about the probability that the CTMDP traverses a certain path. Now, the first problem is that the set of timed paths is uncountable and we must find an appropriate σ-algebra. Given a path length n, we use the product σ-algebra for all paths of length n by combining standard power-set σ-algebras (for states and actions, which are countable) with standard Borel σ-algebras (for the uncountable time-points). This construction is explained in detail by Wolovick and Johr [52] and follows along the same lines as our σ-algebra construction for the paths of an I/O-IMC (see Section 6.2). We say that a set of paths of length n is measurable if it is a member of the product σ-algebra for paths of length n. We find the σ-algebra for all finite timed paths by taking the countable union of the σ-algebras of the paths of a particular length. In general, measurable sets of paths can be constructed by combining states, actions, and time-intervals.

The second problem when assigning probabilities to paths is that CTMDPs are non-deterministic. A *scheduler* resolves this non-determinism by selecting, according to a probability distribution, the action which the CTMDP should take. A *full-history* scheduler bases its decision on the history of the CTMDP, that is the timed path traversed so far.

7.4.1 Early schedulers.

In our treatment of CTMDPs we will consider *early* schedulers which base their decision on the full-history of the CTMDP up to the last state visited, but not on the amount of time the CTMDP has remained in this last state. In other words, an early scheduler decides which action to take immediately upon entering a state and cannot change this decision until the state is left.

Definition 86 ([52]). *A measurable full-history early scheduler for M is a function $D : \mathsf{CPaths} \times A \to [0,1]$ from paths to distributions over the actions A such that for any path σ ending in a state x, we have*

1. *$D(\sigma, \cdot)$ is a probability function for fixed $\sigma \in \mathsf{CPaths}$ and $t \in \mathbb{R}_{\geq 0}$,*
2. *$D(\sigma, a) > 0$ implies that a is enabled in x, and*
3. *D is Borel-measurable, i.e., for any probability $p \in [0,1]$, the set*

$$\{(\sigma, a) \mid D(\sigma, a) = p\}$$

is in the standard Borel σ-algebra over pairs of paths and actions.

Each measurable scheduler yields a probability measure over the timed paths.

Definition 87 ([52]). *Given a measurable full-history early scheduler D, we find for each $n \in \mathbb{N}_0$ a probability measure $\mathcal{P}_D^{(n)}$ on the standard σ-algebra for paths of M of length n. These probabilities measures are defined inductively. For a path of length 0, $\langle x \rangle$, with $x \in S$, we find*

$$\mathcal{P}_D^{(0)}(\langle x \rangle) = \begin{cases} 1, & \text{if } x = x_0, \\ 0, & \text{otherwise.} \end{cases}$$

Given a measurable set of paths H_n of length $n \in \mathbb{N}_0$, let H_{n+1} be a one-step extension of H_n. That is, for an action $a \in A$, time-points $s < u \in \mathbb{R}_{\geq 0}$ and a state $y \in S$ we have

$$H_{n+1} = H_n \times \{a\} \times (s, u] \times \{y\}.$$

Then we find

$$\mathcal{P}_D^{(n+1)}(H_{n+1}) = \int_s^u \int_{\substack{\sigma \in H_n \\ \sigma_t(n) > t}} \mathcal{P}_D^{(n)}(d\sigma) D(\sigma, a) R(\sigma_x(n), a, y) e^{-R(\sigma_x(n), a)(t - \sigma_t(n))} dt.$$

Note that any measurable set of paths of length n must be a countable union of such one-step extensions of paths of length one.

As for I/O-IMCs we assume the time-integral to be a Riemann-integral and the path-integral to be a Lebesgue-integral. That is, $d\sigma$ denotes a set of paths $\{\sigma' \mid D(\sigma', a) = c\}$ for some constant $c \in [0,1]$ and σ is an arbitrary path in $d\sigma$.

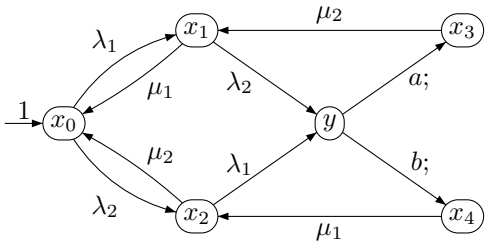


Figure 7.1: Example of a complete I/O-IMC P.

7.4.2 Late schedulers

It should be noted that apart from early schedulers there also exist the class of *late schedulers* for CTMDPs. Late schedulers *can* base their decision on the time at which the last state is left [37], in contrast to early schedulers which must make this decision at the point in time the last state is entered. This means that late schedulers have strictly more information than early schedulers. We will focus our attention on early schedulers in the remainder of the thesis.

7.5 Closed I/O-IMCs and CTMDPs

We now discuss a translation between I/O-IMCs and CTMDPs. We will show that there is a one-to-one correspondence between complete I/O-IMCs and CTMDPs when we consider early schedulers for CTMDPs. This translation is directly derived from the translation of closed IMCs to CTMDPs described by Johr [30].

Example 28. *As a running example we consider the complete I/O-IMC depicted in Figure 7.1 which represents the behaviour of a fault-tolerant system with two-components that may fail (after delays that are exponentially distributed with rates λ_1 respectively λ_2) and can be repaired (after delays that are exponentially distributed with rates μ_1 respectively μ_2). Only one component may be repaired at the same time. When both components are down (represented by state y), the component to be repaired first is selected non-deterministically. Examples of timed paths of the I/O-IMC P of length* 0, 1 *and* 2 *are*

$$\langle x_0, \epsilon, x_0\rangle, \langle x_0, \epsilon, x_0, 3.4, x_1, \epsilon, x_1\rangle, \text{ and } \langle x_0, \epsilon, x_0, 3.4, x_1, \epsilon, x_1, 2.1, y, \epsilon, x_4\rangle.$$

The last path represents the failure of the first component after 3.4 *time-units, followed by the failure of the second component after another* 2.1 *time-units. It is then decided that the first component will be repaired first by moving to state x_4.*

7.5.1 Translation of I/O-IMCs and CTMDPs

We now discuss a translation from I/O-IMCs to CTMDPs that is based on the translation of IMCs to CTMDPs defined by Johr [30]. We consider a closed I/O-IMC

$P = (S, A, R^I, R^M, \alpha)$.

Careful inspection shows some obvious similarities between Definition 87, which describes finite-jump probabilities for a CTMDP with an "early" scheduler and Theorem 44 which describes the finite jump probabilities for a closed I/O-IMC with an interactive jump scheduler (under the assumption that no external jumps occur, i.e., that the induced behaviour is closed). The main problem is the discrepancy between the way we defined timed paths for I/O-IMCs and the definition of timed paths for CTMDPs. However, we will see that this discrepancy is only superficial. Consider the set of all timed paths of length n of I/O-IMC P. We can rearrange this set to find

$$\begin{aligned} Paths^{(n)}_{S,A} &= S_\perp \times \mathcal{L}^V \times S_\perp \times (\mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp)^n \\ &= S_\perp \times (\mathcal{L}^V \times S_\perp \times \mathbb{R}_{\geq 0} \times S_\perp)^n \times \mathcal{L}^V \times S_\perp. \end{aligned}$$

Now, consider a set of states $\bar{S} = S_\perp$ and a set of actions $\bar{A} = \mathcal{L}^V \times S_\perp$ and we find that the above equals

$$\bar{S} \times (\bar{A} \times \mathbb{R}_{\geq 0} \times \bar{S})^n \times \bar{A},$$

which is the set of timed paths for a CTMDP with states $\bar{S}$ and actions $\bar{A}$, extended by a single action from $\bar{A}$.

For the sake of simplicity and to match [30], we consider closed I/O-IMCs

1. whose initial distribution is Dirac,
2. that do not have any absorbing states (i.e., states with no outgoing transitions),
3. that do not contain any interactive cycles containing output transitions, and
4. that do not have any time-divergent states.

The main idea of the following translation is that, when our CTMDP occupies a state x, we may choose non-deterministically one of the fair-reach traces of x. When a fair-reach trace (w, y) is selected, the CTMDP takes on the Markovian behaviour of state y

Recall that S_s denotes the set of stable states and S_u denotes the set of unstable states of an I/O-IMC.

Definition 88 (Adapted from [30])**.** *Given the closed I/O-IMC $P = (S, A, R^M, R^I, \alpha)$ with*

1. *finitely many states and actions,*
2. *no time-divergent or absorbing states,*
3. *no interactive cycles containing output transitions, and*
4. *where α is a Dirac distribution attributing probability one to a state $x_0 \in S$,*

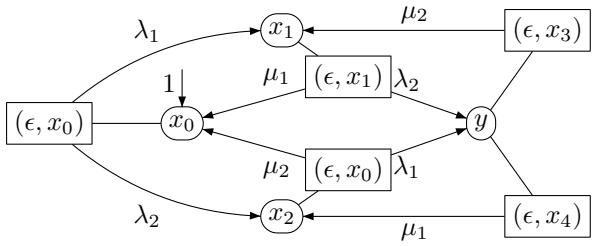


Figure 7.2: Early-CTMDP translation of I/O-IMC P in Figure 7.1. States are denoted as circles, actions as boxes.

we construct the CTMDP $\mathsf{EC}(P) = (S, \bar{A}, R, x_0)$*, with actions*

$$\bar{A} = \cup_{x \in S} \mathit{FairRT}(x)$$

and rate matrix

$$R = \{(x, (w, y), q_{y,z}, z) \mid x \in S, (w, y) \in \mathit{FairRT}(x), z \in S \setminus \{y\}\}.$$

The above is adapted from Johr [30] with a few changes, namely

1. The actions of the CTMDP are named after reach-traces in the associated I/O-IMC instead of traces, and
2. For pairs x, y of stable states such that there is a Markovian transition from x to y, we use the state y as a state in the CTMDP ((ϵ, y) is also the only action enabled in the state y) instead of introducing a new state (x, y) as Johr does [30].
3. We do not consider the reachability of states.

It is important to make sure that Definition 88 always yields a CTMDP. The set of states S is obviously finite. So is the set of actions $\bar{A}$, since the finiteness of A and the lack of interactive cycles with visible transitions means $\mathit{FairRT}(x)$ is finite for each state $x \in S$. For every pair of state x and action (w, y) we have that either $(w, y) \in \mathit{FairRT}(x)$ and then (w, y) is enabled in x and we find for every state $z \in S$ that $|\{R(x, (w, y), z)\}| = 1$ or $(w, y) \notin \mathit{FairRT}(x)$ and then (w, y) is not enabled in x and $|\{R(x, (w, y), z)\}| = 0$.

It is interesting to note that we find the following connection between the rates in $\mathsf{EC}(P)$ and the rates in P, for $x, z \in S$, $y \in S_s$, and $w \in \mathcal{L}^V$,

$$R(x, \langle w, y \rangle, z) = q_{y,z} \text{ and } R(x, \langle w, y \rangle) = q_y. \tag{7.8}$$

Example 29. *Figure 7.2 shows the translation, according to Definition 88, of the I/O-IMC P from Example 28.*

Recall that we made several restrictions on the I/O-IMCs we considered in Definition 88. We will now briefly discuss how we might alleviate these restrictions.

1. We can allow I/O-IMCs with output actions by changing the set of actions for the induced CTMDP to $\mathcal{L}^V \times S_s$. However, this would lead to an infinite set of actions for the CTMDP, although for each state only a finite set of actions may be enabled, and the set of actions might be restricted to only those that are enabled in some state. Note that the set of enabled actions of a state is then simply the set of fair reach-traces.

2. The restriction of the initial distribution to be Dirac may be lifted by changing the definition of CTMDPs to allow initial distributions instead of an initial states. It seems this would not cause any theoretical problems.

3. I/O-IMCs with time-divergent states may be accommodated by adding a distinguished state $\perp$ to the state space of the CTMDP which represents any time-divergent state. However, in combination with allowing visible actions, a state in the I/O-IMC may have infinitely many fair reach-traces (e.g., reach-traces $(\langle a \rangle, \perp), (\langle aa \rangle, \perp), (\langle aaa \rangle, \perp), \ldots$).

We can also translate CTMDPs back to I/O-IMCs. As the state space of such an I/O-IMC we take the union of the states S of the CTMDP and all combinations of states and actions (A) of the CTMDP. The I/O-IMC states of the form $x \in S$ represent the non-deterministic choices made by the CTMDP, while states of the form $(x, a) \in S \times A$ represent the Markovian transitions that are taken *after* the action a was chosen in state x.

Definition 89. *Given a CTMDP $M = (S, A, R, x_0)$, we find the I/O-IMC $\mathsf{IO}(M) = (S \cup S \times A, \{\tau\}, R^M, R^I, \alpha)$ where*

- *the Markovian transition relation is given by*

$$R^M = \{((x, a), \lambda, y) \mid x, y \in S, a \in A_x, R(x, a, y) = \lambda\},$$

- *the interactive transition relation is given by*

$$R^I = \{(y, \tau, (y, a)) \mid y \in S, a \in A_y\},$$

- *and the initial distribution α is a Dirac distribution which assigns probability one to state x_0.*

7.5.2 Translation of schedulers

Consider a closed I/O-IMC $P = (S, A, R^M, R^I, \alpha)$ as in Definition 88 and the corresponding CTMDP $M = \mathsf{EC}(P)$. We now show that there is a one-to-one correspondence between the interactive jump schedulers of P and the full-history schedulers of M. It is important to note that there is a close correspondence between the timed paths of P and the timed paths of M. Given a timed path of length $n \in \mathbb{N}_0$,

$$(x_0, w_0, y_0, t_1, x_1, w_1, y_1, \ldots, t_n, x_n, w_n, y_n),$$

of P we have that for some $t \in \mathbb{R}_{\geq 0}$ and $z \in S$ the sequence

$$(x_0, \langle w_0, y_0 \rangle, t_1, \ldots, x_n)$$

is a timed path of length n of M. Similarly, given a timed path of length $n \in \mathbb{N}_0$,

$$(x_0, \langle w_1, y_1 \rangle, t_1, x_1, \ldots, \langle w_n, y_n \rangle, t_n, x_n)$$

of M we have that for some $w \in \mathcal{L}^V$ and $y \in S$ the sequence

$$(x_0, w_1, y_1, t_1, x_1, w_2, y_2, \ldots, t_n, x_n, w, y),$$

is a timed path of length n of P. We now introduce some notation to facilitate this connection between the timed paths of P and M.

Definition 90. *Given a finite timed path of length n for I/O-IMC P,*

$$\sigma = (x_0, w_0, y_0, t_1, x_1, w_1, y_1, \ldots, t_n, x_n, w_n, y_n),$$

a time-point $t \in \mathbb{R}_{\geq 0}$, and a state $y \in \mathbb{R}_{\geq 0}$, we write – by abuse of notation – $\mathsf{EC}(\sigma)$ for the timed path

$$(x_0, \langle w_0, y_0 \rangle, t_1, \ldots, x_n)$$

of length n of M. We lift this translation to sets of paths. We write

$$\mathsf{EC}(H) = \{\mathsf{EC}(\sigma) \mid \sigma \in H\},$$

for a subset of finite timed paths H of P. Given a finite timed path σ of M of length n, a sequence of actions $w \in \mathcal{L}^V$, and a state $y \in S$, we will write $\mathsf{IO}(\sigma, w, y)$ for the finite timed path σ' of P such that $\mathsf{EC}(\sigma') = \sigma$, $\sigma'_w(n) = w$, and $\sigma'_y(n) = y$. Again, we lift this notation to sets of paths.

Let us consider the set of all finite timed paths of P which are in a sense "reasonable", i.e., all those paths σ where only fair reach-traces are chosen.

Definition 91. *Given an I/O-IMC P and a size $n \in \mathbb{N}_0$, the set of all* finite fair timed paths *of length n, denoted $FFPaths_P^{(n)}$, is the set of all timed paths of P, where only fair-reach traces are chosen. That is,*

$$FFPaths_P^{(n)} = \{\sigma \in Paths_{S,A}^{(n)} \mid \forall 0 \leq i \leq n \cdot (\sigma_w(i), \sigma_y(i)) \in FairRT(\sigma_x(i))\}.$$

The set of all finite fair timed paths is denoted $FFPaths_P$. In other words, we have $FFPaths_P = \bigcup_{n=0}^{\infty} FFPaths_P^{(n)}$.

We now have that for each $\sigma \in \mathsf{CPaths}_M$, $w \in \mathcal{L}^V$, and $y \in S$, such that $(w, y) \in FairRT(last(\sigma))$, there exists exactly one path $\sigma' \in FFPaths_P$ such that $\mathsf{EC}(\sigma') = \sigma$ and w, y are the last two entries of σ'. In a sense, the function EC is a bijection from $FFPaths_P$ to $\mathsf{CPaths} \times \mathcal{L}^V \times S$, if we only consider the finite fair timed paths. Based on this correspondence between the timed paths of I/O-IMCs and their CTMDP counterparts, we now define a translation from interactive jump schedulers of the I/O-IMC P to full-history measurable schedulers of the associated CTMDP M.

Definition 92. *Given an interactive jump scheduler γ for P we find the measurable full-history early scheduler $D = f_E(\gamma)$ for M. We first consider timed paths of length zero. Given states $x, y \in S$ and a sequence of actions $w \in \mathcal{L}^V$ we find for the probability of choosing action (w, y) after path (x), that*

$$D(\langle x \rangle, \langle w, y \rangle) = \gamma^{(0)}_{\epsilon,x}(w, y). \tag{7.9}$$

Now, consider a timed path σ of M of length $n > 0$. We find for the probability of choosing action (w, y) after the path σ, that

$$D(\sigma, \langle w, z \rangle) = \gamma^{(t)}_{\sigma',x}(w, y), \tag{7.10}$$

where σ' is the unique path of length $n-1$ of P such that $\mathsf{EC}(\sigma' \circ \langle t, x, w, y \rangle) = \sigma$. Conversely, given a measurable full-history early scheduler D for Q, we find the interactive jump scheduler $\gamma = f_I(D)$ for P, which is also given by (7.9) and (7.10), where now the left-hand side is given and the right-hand side defines γ.

Proposition 26. *For any interactive jump scheduler γ of P, $f_E(\gamma)$ is indeed a full-history measurable scheduler of M, and $f_I(f_E(\gamma)) = \gamma$. Similarly, for any full-history measurable scheduler D of M, we have that $f_I(D)$ is indeed an interactive jump scheduler of P, and $f_E(f_I(D)) = D$.*

The proof of Proposition 26 can be found in Appendix A.2.5. We now show that the translations between I/O-IMC schedulers and early CTMDP schedulers preserve the induced finite-jump probabilities.

Theorem 50. *For any interactive jump scheduler γ for P, which induces a closed behaviour X with history process Z, and its counterpart $D = f_E(\gamma)$ for M, we have that*

1. *for a state $x \in S$*

$$\Pr(Z^{(J_0)} \in \{x\} \times \mathit{FairRT}(x)) = \mathcal{P}^{(0)}_D(\{x\}), \text{ and} \tag{7.11}$$

2. *given a measurable set of finite fair timed paths of length $n \in \mathbb{N}_0$*

$$H_n = \{(x_0, w_0, y_0)\} \times (s_1, u_1] \times \{(x_1, w_1, y_1)\} \times \ldots \times (s_n, u_n] \times \{(x_n, w_n, y_n\},$$

with states $x_0, y_0, \ldots, x_n, y_n \in S$, sequences of actions $w_0, \ldots, w_n \in \mathcal{L}^V$, time-points $s_1, u_1, \ldots, s_n, u_n \in \mathbb{R}_{\geq 0}$, such that $(w_i, y_i) \in \mathit{FairRT}(x_i)$ for all $0 \leq i < n$, and $y_i \neq x_{i+1}$ for all $0 \leq i \leq n$, we have for time-points $s, u \in \mathbb{R}_{\geq 0}$, a state $x \in S \setminus \{y_n\}$, that

$$\Pr(Z^{(J_{n+1})} \in H_n \times (s, u] \times \{x\} \times \mathit{FairRT}(x)) = \mathcal{P}^{(n+1)}_D(\mathsf{EC}(H_n \times (s, u] \times \{x\} \times \mathit{FairRT}(x))). \tag{7.12}$$

The proof of Theorem 50 can be found in Appendix A.2.6.

We have shown in this section that there is a one-to-one correspondence between closed I/O-IMCs and CTMDPs. This is not surprising given the work of Johr which shows a similar correspondence between closed IMCs and CTMDPs [30]. In the case of I/O-IMCs, however, the correspondence is not presented as a monolithic translation, but arises naturally from the semantics given to *open* I/O-IMCs in Chapter 6 and the rules for composing these interactive jump processes in parallel.

7.6 Closed behaviours of deterministic I/O-IMCs

We now show several results for deterministic and confluent I/O-IMCs. In particular, we will show that if an I/O-IMC is weakly deterministic, it indeed has only a single scheduler. Recall that a closed I/O-IMC P is weakly deterministic if, for any stochastically reachable state x of P we have that the IOA rooted at x is weakly deterministic, i.e., its outgoing internal transitions go to weakly bisimilar states and it has no choices between different visible actions (cf. Definition 45). One of the important results we showed is that the weak bisimulation quotient of a weakly deterministic I/O-IMC always has no internal transitions (except for self-loops) and each state has at most one outgoing interactive transition. As a consequence each state of this quotient has only one non-divergent fair reach-trace.

Theorem 51. *Consider a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ with no internal transitions and where for each state $x \in S$ we have*

$$x \xrightarrow{a} y, x \xrightarrow{b} z \text{ implies } a = b, y = z.$$

For any closed behaviour X of P that is non-explosive, i.e.,

$$\Pr(J_\infty = \infty) = 1,$$

we have that X_{post} is a Markov chain.

The proof of Theorem 51 can be found in Appendix A.2.7. We can now show that weakly deterministic I/O-IMCs are Markovian, at least if we only observe which weak bisimulation equivalence class they occupy.

Theorem 52. *Given a complete, weakly deterministic I/O-IMC P, for any closed behaviour X of P that is non-explosive, i.e.,*

$$\Pr(J_\infty = \infty) = 1,$$

we have that the stochastic process

$$Y^{(t)} = \left[X^{(t)}_{\mathsf{post}}\right]_{\approx},$$

which records which weak bisimulation equivalence class is occupied by X_{post}, is a Markov chain.

Proof. The I/O-IMC P is weakly bisimilar to its weak bisimulation quotient, which satisfies the conditions of Theorem 51 due to Proposition 15. Now, Theorem 51 gives us that all closed, non-explosive behaviours of the weak bisimulation quotient of P yield the same Markov chain for X_{post}. In essence, $[P]_{\approx}$ has only a single possible closed behaviour. Now, Theorem 47 tells us that each closed behaviour of P can be simulated by a closed behaviour of $[P]_{\approx}$ (with respect to the weak bisimulation equivalence classes). It immediately follows that Y_{post} is indeed a Markov chain, namely the same Markov chain we find for $[P]_{\approx}$. □

Note that the jump process X_{post} of a closed behaviour of a weakly deterministic I/O-IMC need not be a Markov chain as the probability to be in a particular state of an equivalence class may depend on the past, even though the probability to occupy *any* state in the equivalence class has the Markov property (and Y_{post} is a Markov chain).

7.7 Discussion

In this chapter we have shown that – in general – an I/O-IMC that does not interact with its environment corresponds to a CTMDP, and – specifically – a *deterministic* I/O-IMC that does not interact has as its semantics a CTMC. These results closely match the results of Johr for IMCs [30] and indeed our translation from I/O-IMCs to CTMDPs is heavily inspired by Johr's. This is not surprising as, for closed models, the distinction between I/O-IMCs and IMCs disappears since this distinction only pertains to the way I/O-IMCs and IMCs interact.

However, there is a crucial difference between the result in this chapter and Johr's result. Whereas in Johr's case, the translation from (closed) IMC to CTMDP gives a monolithic semantics to IMCs, in our case this translation arises from the non-monolithic, modular semantics introduced in Chapter 6. This also proves that the translational semantics derived from Johr coincides with the modular semantics from Chapter 6.

7.7.1 Markovian schedulers

In general, the closed behaviours that we can derive from non-deterministic closed I/O-IMCs are not Markov chains, i.e., they do not satisfy the Markov property. This is caused by the fact that the decision, which fair-reach trace to choose, may depend on the history of the behaviour. However, certain behaviours of non-deterministic closed I/O-IMCs do correspond to Markov chains. Without proof we note that these behaviours are exactly the behaviours induced by so-called *Markovian schedulers*, i.e., schedulers where the decision, which fair reach-trace to take, depends only on the current state of the behaviour, not on the past. Such schedulers satisfy

$$\gamma^{(t)}_{\sigma,x}(w,y) = \gamma^{(t')}_{\sigma',x}(w,y)$$

for any pair of times $t, t' \in \mathbb{R}_{\geq 0}$ and pair of paths σ, σ'. This is not surprising as the same is the case for CTMDPs.

7.7.2 Analysis

Because CTMDPs are non-deterministic, it is not possible to compute the probability for a CTMDP to occupy a particular state at a particular time. However, it is possible to compute infima and suprema for such transient probabilities considering all different schedulers for the CTMDP. We will consider the problem of computing the infimum and supremum probability of occupying a particular set of states at time t for a CTMDP with m transitions and a maximal exit-rate of λ. For general CTMDPs, Neuhäusser

and Zhang gave an algorithm that computes these infima and suprema to within a predetermined error-bound ϵ, which has time-complexity $O(m(\lambda t)^2\epsilon^{-1})$ [38]. Note that λt is an upper-bound for the expected number of jumps to occur within time t.

A very popular way of computing the transient distribution for CTMCs is to use *uniformisation* [29]. Given a CTMC with m transitions (i.e., m non-zero entries in its infinitesimal generator matrix), we can use uniformisation to compute the transient probability of the CTMC at time t with a time-complexity of $O(m\lambda t)$ where λ is the maximal exit-rate appearing in the CTMC.

There are many other ways of analysing CTMCs. We would like to mention two of these in particular, as – in contrast to uniformisation – these solution techniques do not require the construction of the entire state space of the CTMC. Simulation [19] can be used to estimate the transient distribution of a CTMC by generating "runs" of the CTMCs and using statistical methods. Fast adaptive uniformisation [15], is a variant of uniformisation, where – at different iterations of the algorithm – only a subset of significant states is maintained instead of considering the entire infinitesimal generator matrix. Crucially, the complexity of such *on-the-fly* analysis methods does not depend on the size of the state space of the CTMC, but rather on the shape of its transient distributions. In Chapter 9 we will consider the possibility of applying on-the-fly analysis methods directly on a composition of I/O-IMCs without building its state space. However, in order to do this we must first know whether such a composite I/O-IMC is deterministic or not. The question of how to determine whether or not a composite I/O-IMC is deterministic – without building its state space – is one of the topics of the next chapter.

8 Determinism

In Chapter 7 we have seen that whether or not an I/O-IMC is deterministic, makes a huge difference in its interpretation. Deterministic I/O-IMCs have as their semantics a single interactive jump process, which in turn describes a CTMC. To analyse a deterministic I/O-IMC we can then analyse this underlying CTMC. On the other hand, non-deterministic I/O-IMCs do not correspond to CTMCs, but rather to CTMDPs. This means that the analysis of non-deterministic I/O-IMCs is significantly more complex than the analysis of deterministic I/O-IMCs and can only yield bounds for the quantitative properties of an I/O-IMC instead of exact quantities as is the case for deterministic I/O-IMCs. It is then interesting to know whether an I/O-IMC is deterministic or not. This is the question we study in this chapter. We further study the question whether an I/O-IMC is divergent or not.

For a composite I/O-IMC $\mathcal{C} = (P_1 \| \dots \| P_n) \backslash A^H$ it is generally necessary to construct and examine the state space and transitions of $\mathcal{C}$ explicitly to determine whether it is deterministic and/or time-divergent. The size of the state space of course grows exponentially in the number of components, which means that constructing this state space may be infeasible or at least impractical. In Chapter 9 we will see that we can analyse a composite I/O-IMC without constructing its entire state space, if we know that it is deterministic and non-divergent. For this reason it would be extremely useful to be able to efficiently show that certain I/O-IMCs are indeed deterministic and non-divergent.

Contribution. In this chapter we will develop sufficient conditions that ensure that a complete composite I/O-IMC is deterministic. These conditions can be checked in *polynomial* time and space with respect to the size of the *component* models of the composite I/O-IMC. We will also develop similar sufficient conditions that ensure that a complete composite I/O-IMC is non-divergent. In our examples and explanations we

will first focus on the issue of determinism, and later show how we can also apply the proposed methods to divergence.

8.1 Confluence and reachability

In Chapter 5 we have discussed the notion of weak confluence for I/O-IMCs that is preserved by the architectural operations on I/O-IMCs. This means that a composite I/O-IMC is weakly confluent if its components are weakly confluent. This immediately gives us an efficient overapproximation of the weak confluence of a composite I/O-IMC by simply checking the weak confluence of its components. However, this method may lead to many spurious counter-examples, i.e., there are many confluent composite I/O-IMCs which are composed of non-confluent I/O-IMCs. One of the reasons for this is that stochastic reachability is not preserved by parallel composition. In this chapter we will develop means to eliminate spurious counterexamples when checking if a composite I/O-IMC is weakly confluent. In particular, we concentrate on what causes actions to become enabled in an I/O-IMC.

Example 30. *Consider a closed composite I/O-IMC* $C = (P_1 \| P_2 \| P_3) \setminus \{a, b\}$*, whose components are shown in Figure 8.1. We have output actions* $A_1^O = \{a\}$*,* $A_2^O = \{b\}$*, and* $A_3^O = \{c\}$*, input actions* $A_1^I = A_2^I = \emptyset$ *and* $A_3^I = \{a, b\}$*, and no internal actions. We can*

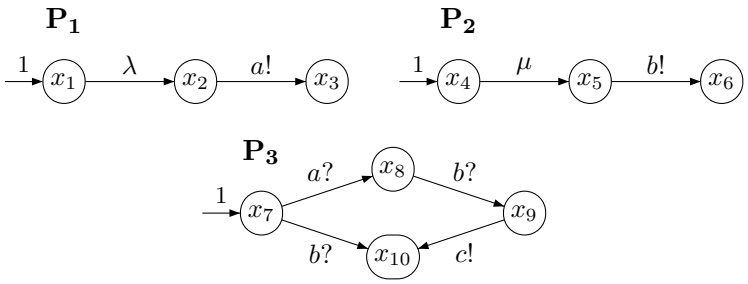


Figure 8.1: The components of closed composite I/O-IMC C. Self-loops labelled with input actions have been left out for the sake of simplicity.

see that, although P_1 *and* P_2 *are weakly confluent,* P_3 *is not. Condition* (4.11) *is violated in state* x_7 *of I/O-IMC* P_3 *because the sequence ab leads to a different state (*x_9*) than the sequence ba (which leads to* x_{10}*). The I/O-IMC C is depicted in Figure 8.2. The equivalence classes of* $\approx_C$*, omitting equivalence classes which consist of a single state, are drawn as dashed boxes. We can easily see that all states except state* $x_2 \| x_5 \| x_7$ *satisfy the conditions of weak determinism. State* $x_2 \| x_5 \| x_7$ *on the other hand obviously has outgoing internal transitions to states that are not weakly bisimilar. However, the state is in fact not stochastically reachable. For the two paths* $x_1 \| x_4 \| x_7 \xhookrightarrow{\lambda} x_2 \| x_4 \| x_7 \xhookrightarrow{\mu} x_2 \| x_5 \| x_7$ *and* $x_1 \| x_4 \| x_7 \xhookrightarrow{\mu} x_1 \| x_5 \| x_7 \xhookrightarrow{\lambda} x_2 \| x_5 \| x_7$ *that lead to state* $x_2 \| x_5 \| x_7$ *we find that they are not plausible as both states* $x_2 \| x_4 \| x_7$ *and* $x_1 \| x_5 \| x_7$ *are unstable.*

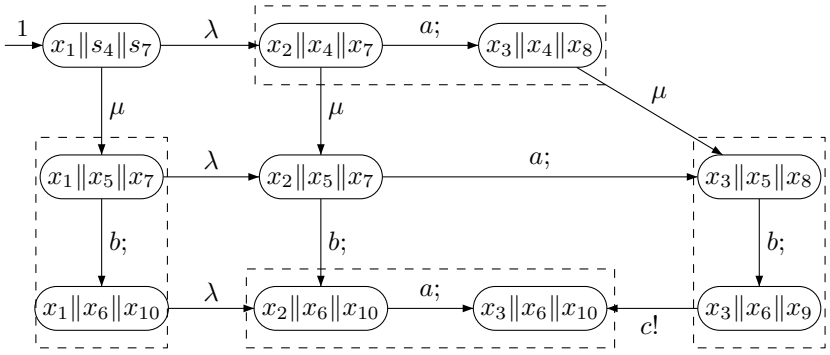


Figure 8.2: The states, transitions, and weak bisimulation equivalence classes of closed composite I/O-IMC C.

In the above example we have that the violation of condition (4.11) by the third component of composite I/O-IMC C is spurious, because the one state of C where the actions a and b are both enabled is not stochastically reachable. In this chapter we will develop an efficient way to prove that certain actions cannot be enabled at the same time, or more precisely: that the probability that these actions are enabled at the same time is zero. We will also show that, if two actions a and b cannot be enabled at the same time, condition (4.11) can be safely ignored for this pair of actions.

In general, determining the (stochastic) reachability of a state in a transition system involves building the state space of the transition system. Of course, it is undesirable to generate the state space of a composite I/O-IMC as this state space grows exponentially in the number of components. We therefore develop a way to safely approximate which actions can be enabled at the same time by considering which events directly cause actions to become enabled. We will see that for composite I/O-IMCs this *causal* relationship can be closely approximated by considering only the component I/O-IMCs.

8.2 Spontaneously enabled actions

We say a set of actions is *spontaneously enabled* if they may become enable through a random event. In the context of an I/O-IMC a set of actions is spontaneous if there is a state in the I/O-IMC, where all the actions are enabled and that state can be reached via a plausible Markovian transition.

Definition 93 (Spontaneous actions)**.** *Given an I/O-IMC P with state space S and actions A_P, we say a set of locally-controlled actions $B \subset A_P^O \cup A_P^H$ is* spontaneous *if there are stochastically reachable states x and x' in S such that*

- *x is stable,*
- *there is a Markovian transition from x to x', i.e., $x \xhookrightarrow{\lambda} x'$ for some $\lambda \in \mathbb{R}_{\geq 0}$, and*

- *all actions in B are enabled in x', i.e., $\forall b \in B \cdot (\exists x'' \in S \cdot x' \xrightarrow{b} x'')$.*

Given an I/O-IMC P, we say that a set of actions B is maximally spontaneous if B is spontaneous in P and there exists no strict superset B' of B that is also spontaneous in P. For convenience, we will say an action b is (maximally) spontaneous if the singleton set $\{b\}$ is (maximally) spontaneous.

In Example 30 we have that actions a and b are both spontaneous in P_1 and P_2, respectively. However, action c is not spontaneous, as the only state in which c is enabled, state s_9, is not reachable with a Markovian transition. For the composite I/O-IMC C from the same example we see that the actions a and b are also spontaneous. It is unfortunately not the case that spontaneity is preserved by parallel composition, however we find that *non-spontaneity* is indeed preserved by parallel composition and hiding.

Lemma 19. *Given compatible I/O-IMCs P_1 and P_2, we have that if a set of actions B is spontaneous in $P_1 \| P_2$, then B is spontaneous in either P_1 or P_2.*

Proof. We prove Lemma 19 by contradiction. Let A_1 and A_2 be the actions for I/O-IMCs P_1 and P_2 respectively. Assume then that there exists a set of actions $B \subset A_1^O \cup A_2^O \cup A_1^H \cup A_2^H$ such that B is spontaneous in $P_1 \| P_2$, but B is not spontaneous in P_1 nor in P_2. We will show that this assumption leads to a contradiction.

Since B is spontaneous in $P_1 \| P_2$ we find stochastically reachable states $x_1 \| x_2$ and $x_1' \| x_2'$ in $P_1 \| P_2$ such that $x_1 \| x_2$ is stable, $x_1 \| x_2 \xhookrightarrow{\lambda} x_1' \| x_2'$, and for each action b in B we find a state $x_1'' \| x_2''$ such that $x_1' \| x_2' \xrightarrow{b} x_1'' \| x_2''$. The following then follow from our modularity results for parallel composition:

1. x_1 and x_2 are stable in P_1 respectively P_2,
2. there is either a transition $x_1 \xhookrightarrow{\lambda} x_1'$ in P_1 with $x_2 = x_2'$ or a transition $x_2 \xhookrightarrow{\lambda} x_2'$ in P_2 with $x_1 = x_1'$,
3. for each action $b \in B \cap (A_1^O \cup A_1^H)$ controlled by P_1 we find a state x_1'' for P_1 and a transition $x_1' \xrightarrow{b} x_1''$, and
4. for each action $b \in B \cap (A_2^O \cup A_2^H)$ controlled by P_2 we find a state x_2'' for P_2 and a transition $x_2' \xrightarrow{b} x_2''$.

From Theorem 33, we furthermore have that x_1 and x_1' are stochastically reachable in P_1 and x_2 and x_2' are stochastically reachable in P_2. We now consider whether P_1 or P_2 control the actions in B.

First consider the case that the actions in B are all controlled by P_1, i.e., $B \subset (A_1^O \cup A_1^H)$. We then immediately have that B is spontaneous in P_1 which constitutes a contradiction. We find a similar result for the case that the actions in B are controlled by P_2.

Let $B_1 = B \cap (A_1^O \cup A_1^H)$ and $B_2 = B \cap (A_2^O \cup A_2^H)$ be sets of actions of B controlled by P_1 respectively P_2. Now consider the case that B_1 and B_2 are both non-empty. We

then have that both x'_1 and x'_2 are unstable, because of items 3 and 4 above. However, we also have that either x_1 is equal to x'_1 or x_2 is equal to x'_2, because of item 2. It then follows that either x_1 or x_2 is unstable. This is a contradiction with item 1. □

Lemma 20. *Given an I/O-IMC P with actions A and a set of actions $B \subset A^O$, a set of actions $B' \subset A^O \cup A^H$ is spontaneous in P if and only if B' is spontaneous in $P\backslash B$.*

Proof. The proof of Lemma 20 is straightforward, as hiding does not affect stability or stochastic reachability. □

Theorem 53. *Given a composite I/O-IMC $C = (P_1 \| \dots \| P_n)\backslash A^H_C$, if a set of actions $B \subset A^O_C \cup A^H_C$ is spontaneous in C then it is spontaneous in one of its components.*

Proof. Theorem 53 follows directly from Lemmas 19 and 20. □

Theorem 53 allows us to overapproximate the sets of spontaneously enabled actions in a composite I/O-IMC $C = (P_1 \| \dots \| P_n)\backslash A^H_C$ by considering the sets of spontaneously enabled actions in its components. In particular we have:

$$\{B \mid B \text{ spontaneous in } C\} \subseteq \bigcup_{i=1}^{n} \{B \mid B \text{ spontaneous in } P_i\}. \tag{8.1}$$

We find the same result for the maximally spontaneous sets of actions. This result is promising, but we will see that random events are not the only way in which actions may become enabled.

8.3 Initially enabled actions

We consider a variation of Example 30 to illustrate that sets of actions may become enabled without being spontaneous.

Example 31. *Consider a closed composite I/O-IMC $C' = (P'_1 \| P'_2 \| P_3)\backslash\{a, b\}$ where P'_1 and P'_2 are shown in Figure 8.3 and P_3 is shown in Figure 8.1. The action signatures of the components of C' are identical to those of the composite I/O-IMC C from Example 30. We see that again P'_1 and P'_2 are confluent, while P_3 is not. However, in this*

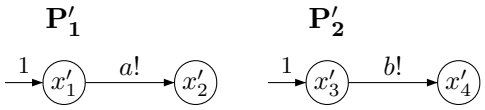


Figure 8.3: Two components of closed composite I/O-IMC C'.

example we have that none of the actions are spontaneous. Figure 8.4 now shows the states, transitions, and weak bisimilarity equivalence classes of C' itself. We see that C' is not weakly deterministic as the state $x'_1 \| x'_3 \| x_7$ is stochastically reachable and has outgoing internal transitions to states that are not weakly bisimilar.

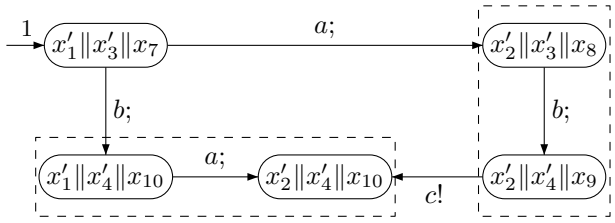


Figure 8.4: The states, transitions, and weak bisimulation equivalence classes of closed composite I/O-IMC C'.

In the above example we have that the actions a and b are *initially* enabled. That is, they are enabled at time-point zero, before any events have occurred.

Definition 94 (Initial actions)**.** *Given an I/O-IMC P with states S, actions A_P, and initial distribution α_P, a set of actions $B \subset A_P^O \cup A_P^H$ controlled by P is* initial *if there is a state x in S_P such that:*

- *x is initial, i.e., $\alpha_P(x) > 0$ and*
- *each action b in B is enabled in x, i.e., for all actions b in B there exists a state $x' \in S_P$ such that there is a transition $x \xrightarrow{b} x'$.*

Given an I/O-IMC P, we say that a set of states B is maximally initial if B is initial in P and there exists no strict superset B' of B that is also initial in P. For convenience, we will say an action b is (maximally) initial if the singleton set $\{b\}$ is (maximally) initial.

In Example 31 both actions a and b are initial. Note that, although action c may occur at time-point zero for I/O-IMC C', it is not initial, because a c-event must be preceded by both an a and a b-event. For the composite I/O-IMC C' we see that the set $\{a, b\}$ is initial. In fact, initiality is additive with respect to parallel composition.

Theorem 54. *Given a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \setminus A_C^H$, we have that a set of actions B is initial for C if and only if we find initial sets $B_1, \dots, B_n$ in the respective components $P_1, \dots, P_n$ of C such that $B = \bigcup_{i=1}^{n} B_i$.*

Proof. Theorem 54 follows directly from the definitions of parallel composition and initially enabled actions. □

Theorem 54 allows us to identify the sets of initially enabled actions in a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \setminus A_C^H$ by considering the sets of initially enabled actions in its components. In particular we have:

$$\{B \mid B \text{ initial in } C\} = \left\{ \bigcup_{i=1}^{n} B_i \mid \forall 1 \leq i \leq n \cdot B_i \text{ initial in } P_i \right\}. \tag{8.2}$$

We find the same result for the maximally initial sets of actions. We now move on to the third and final way in which actions may become enabled in a composite I/O-IMC.

8.4 The triggering relation

We again modify our running example slightly.

Example 32. *Consider a closed composite I/O-IMC* $C'' = (P_1''\|P_2''\|P_3)\backslash\{a, b, d\}$*, where* P_1'' *and* P_2'' *are shown in Figure 8.5 and* P_3 *is shown in Figure 8.1. The first component of* C *now has input action* $A_{1''}^I = \{d\}$ *and the second component now has output actions* $A_{2''}^O = \{b, d\}$*. We once again have that* P_1'' *and* P_2'' *are confluent, and* P_3 *is not. The*

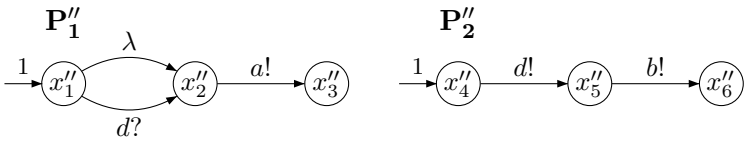


Figure 8.5: Two components of closed composite I/O-IMC C''.

I/O-IMC C'' *is depicted in Figure 8.2. The equivalence classes of* $\approx_{C''}$ *are drawn as dashed boxes. It is obvious that* C'' *is not weakly deterministic.*

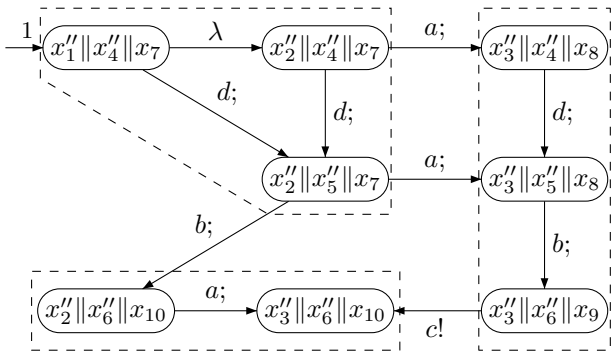


Figure 8.6: The states, transitions, and weak bisimulation equivalence classes of closed composite I/O-IMC C''.

In the above example, the actions a and b are enabled at the same time in state $x_2''\|x_5''\|x_7$ and this state is indeed stochastically reachable. However, only action d is initially enabled and no actions are would be spontaneously enabled (note that state $x_2''\|x_4''\|x_7$ is not stochastically reachable). If we look at the transitions of C'' we can see that first a d-event occurs and then both a and b become enabled. If we look at the components that control the actions a and b (P_1'' and P_2'' respectively) we see that there are transitions labelled d which *enable* a and b, respectively. For P_1'' we see that the transition $x_1'' \xrightarrow{d?} x_2''$ goes from a state where a is not enabled to a state where action a

is enabled. We say that d *triggers* a. Similarly we find for I/O-IMC P_2'' that d *triggers* b.

Definition 95 (Triggering relation)**.** *Given an I/O-IMC P with states S and actions A, for distinct actions $a \in A$ and $b \in A^O \cup A^H$, we say that a* triggers *b in P if there exist stochastically reachable states x_1, x_2, and x_3 in S such that we have:*

$$x_1 \xrightarrow{a} x_2, x_2 \xrightarrow{b} x_3, \text{ and } \nexists x_4 \in S \cdot x_1 \xrightarrow{b} x_4. \tag{8.3}$$

For an action $b \in A^O \cup A^H$ we have that b triggers *b if we find stochastically reachable states x_1, x_2, and x_3 in S such that we have:*

$$x_1 \xrightarrow{b} x_2 \text{ and } x_2 \xrightarrow{b} x_3. \tag{8.4}$$

The triggering relation describes which interactive events may cause interactive events labelled with a particular action to happen. We saw that d triggers a and b in the I/O-IMCs P_1'' and P_2'' respectively, but d also triggers a and b in their parallel composition C''. In fact, the triggering relations of the components of a composite I/O-IMC *overapproximate* the triggering relation of the composite I/O-IMC itself.

Theorem 55. *Given a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \setminus A_C^H$ and two actions $a \in A_C$ and $b \in A_C^O \cup A_C^H$, let P_i be the component of C that controls b. We have that if a triggers b in C then a triggers b in P_i.*

Proof. Theorem 55 easily follows from the definitions of parallel composition and the triggering relation. □

It follows that we can indeed overapproximate the triggering relation of a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \setminus A_C^H$ by combining the triggering relations of its components:

$$\{(a, b) \mid a \text{ triggers } b \text{ in } C\} \subseteq \bigcup_{i=1}^{n} \{(a, b) \mid a \text{ triggers } b \text{ in } P_i\} \tag{8.5}$$

For this reason we will call the union of component triggering relations for C, the *approximate triggering relation* of C.

Definition 96. *Given a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \setminus A_C^H$ the* approximate triggering relation *of C is the union of the triggering relations of the components of C.*

We have shown three different causes of interactive events in I/O-IMCs. We have also shown that for composite I/O-IMCs we can overapproximate these causal relationships by studying only the components of the composite I/O-IMC. In the next section we show that these are indeed the only three causes of interactive events in I/O-IMCs and that we can use the causal relationships to overapproximate which actions may be enabled at the same time in a composite I/O-IMC.

8.5 Enabled sets

To determine whether a composite I/O-IMC is deterministic, it is useful to know which sets of actions may be enabled simultaneously. We call such a set of actions an *enabled set*.

Definition 97 (Enabled sets)**.** *Given an I/O-IMC P with states S and actions A, a set of actions $B \subset A^O \cup A^H$ controlled by P is an* enabled set *of P if there exists a stochastically reachable state x in S, where all actions in B are enabled:*

$$\exists x \in S \cdot SR(x), \forall b \in B \cdot (\exists x' \in S \cdot x \xrightarrow{b} x').$$

We denote the set of all enabled sets of an I/O-IMC P as ES_P.

We first consider single actions that are enabled in an I/O-IMC. The following theorem states that such enabled actions must always be either spontaneous, initial, or triggered by some action.

Theorem 56. *Given an I/O-IMC P, if an action a is enabled in a stochastically reachable state of P, then a is spontaneous, initial, or triggered by an action b.*

Proof. Let x be a stochastically reachable state in P such that action a is enabled in x. This means there exists a state x' in P such that there is a transition $x \xrightarrow{a} x'$. Because x is stochastically reachable there must exist a plausible path σ from an initial state in P to x. We now prove Theorem 56 by induction on the length of σ. As our induction assumption we assume that for all plausible paths σ' that are shorter than σ, start in an initial state, and that lead to a state x'' where an action b is enabled, we have that b is spontaneous, initial, or triggered by some action c.

We consider the nature of the plausible path σ.

- In the case that σ has length zero it follows that x must be an initial state of P and then a is initial.
- In the case that σ has length greater than zero and the last transition of σ is Markovian we have that a is spontaneous.
- Consider now the case that σ has length greater than zero and the last transition of σ is an interactive transition labelled with action b. We find a state x'' in P and a plausible path σ' such that $\sigma = \sigma' \circ x'' \xrightarrow{b} x$. For the state x'' there are two possibilities. First, a may not be enabled in x'' and then b triggers a. Secondly, a may be enabled in x''. Then we find a state x''' such that $x'' \xrightarrow{a} x'''$. Since there is a plausible path to state x'' which has one fewer transition than σ, we can apply the induction assumption and find that a is either spontaneous, initial, or triggered by some action.

This completes the proof of Theorem 56. □

It of course follows that, for an enabled set of actions B in an I/O-IMC we have that each of the actions in B must be either spontaneous, initial or triggered. We will now try to approximate the enabled sets of an I/O-IMC by considering only which actions are spontaneous and initial and which actions trigger other actions. We make the following intuitive observations:

- All spontaneous sets of an I/O-IMC are enabled sets,
- All initial sets of an I/O-IMC are enabled sets, and
- Whenever a set of actions B is enabled and a transition labelled $b \in B$ occurs, we have that b is no longer enabled, but all the actions triggered by b may become enabled.

Note that, for the last item it may be the case that the action b indeed stays enabled because it triggers itself. The first two items above are exact, but the last item is an overapproximation. It does not consider the fact that, the b-transition may disable some of the actions in B, nor does it consider the fact that perhaps not all actions triggered by b are enabled by that particular b-transition. Still, we can use the above observations to overapproximate the enabled sets of a closed I/O-IMC.

Definition 98. *Given a closed I/O-IMC P, the* enabled graph *of P, written EG_P, approximates the sets of actions controlled by P that may be enabled at the same time. It has vertices $V = \bigcup_{i=0}^{\infty} V_i \subset 2^{A_P^O \cup A_P^H}$, labels A, and edges $E = \bigcup_{i=0}^{\infty} E_i$, which are defined recursively for all $i \in \mathbb{N}$:*

$$\begin{aligned} V_0 =& \{B \mid B \text{ is maximally spontaneous}\} \cup \\ & \{B \mid B \text{ is maximally initial}\} \\ E_i =& \{(v, a, (v \setminus \{a\}) \cup \{b \mid a \text{ triggers } b\}) \mid v \in V_i, a \in v\} \\ V_{i+1} =& \{v' \mid v \in V_i, a \in A, (v, a, v') \in E_i, v' \notin \bigcup_{j=0}^{i} V_j\} \end{aligned}$$

We denote the subset-closure of the vertices of the enabled graph (V, E) of P as $\bar{ES}_P$:

$$\bar{ES}_P = \{v' \mid v' \subseteq v, v \in V\}$$

The vertices of the enabled graph of a closed I/O-IMC overapproximate the enabled sets.

Theorem 57. *Given a closed I/O-IMC P, for any enabled set B of P we find that there is a superset of B that is a vertex in the enabled graph of P:*

$$ES_P \subseteq \bar{ES}_P.$$

Proof. Let B be an enabled set of P, we will prove that there is a superset v of B that is a vertex in the enabled graph of P. From the definition of enabled sets, we have that

there is a stochastically reachable state x of P where all the actions in B are enabled. Since x is stochastically reachable we have that there is a plausible path σ from an initial state of P to x. We prove Theorem 57 by induction on the length of σ. As our induction assumption, we assume that for any plausible path σ' that starts in an initial state in P, ends in a state x' where the set of actions B' is enabled, and that is shorter than σ, we have that there exists a superset v' of B' that is a vertex in the enabled graph of P.

We consider the nature of the path σ.

- Consider the case that σ has length zero. Then we have that the state x is an initial state. It immediately follows that there is a maximally initial set of actions v that is a superset of B. This set v must be in V_0 and is thus a vertex of the enabled graph of P.

- Consider the case that σ has length greater than zero and the last transition of σ is a Markovian transition. It then immediately follows that there exists a maximally spontaneous set of actions v that is a superset of B. This set v must again be in V_0 and is thus a vertex of the enabled graph of P.

- Finally we consider the case that σ has length greater than zero and its last transition is an interactive transition labelled a. This means we find a path σ' and a state x' such that $\sigma = \sigma' \circ x' \xrightarrow{a} x$. It is obvious that σ' is plausible, starts in an initial state and is shorter than σ. For each action b in B we either have that b is also enabled in x' or not. If b is not enabled in x' then, by (8.3), a triggers b.

  Let B' be the set containing a and all actions in B that are enabled in s'.

  $$B' = \{a\} \cup \{b \mid b \in B, b \text{ enabled in } x'\}.$$

  Note that a is also enabled in x'. By the induction assumption we find that there is a vertex v' in the enabled graph of P that is a superset of B'. From the definition of the enabled graph we now find a transition (v', v) in the enabled graph where

  $$v = v' \setminus \{a\} \cup \{c \mid a \text{ triggers } c\}.$$

  If a is in B, then a is enabled in x and then, by (8.4), we have that a triggers a. It follows that a is also in v. For an action $b \in B$, with $b \neq a$, we have that, if b is not enabled in x' then a triggers b and b must be in v. For an action $b \in B$, with $b \neq a$, that is enabled in x' we have that b is in v' and then also in v. This shows that each of the actions in B is also in v and then v is a superset of B.

This completes the proof of Theorem 57. □

Determining the spontaneous sets and the triggering relation for a composite I/O-IMC involves generating its state space, which grows exponentially in the number of its components and which may be prohibitively large. However, we have seen throughout this chapter that these causal relations can be approximated by considering the components of the composite I/O-IMC. This naturally leads to an approximate version of the enabled graph.

Definition 99. *Given a closed composite I/O-IMC* $C = (P_1 \| \ldots \| P_n) \backslash A_C^H$*, the* approximate enabled graph *of* C*, written* $\bar{EG}_C$*, approximates the sets of actions controlled by* C *that may be enabled at the same time. It has vertices* $V = \bigcup_{i=0}^{\infty} V_i \subset 2^{A_C^O \cup A_C^H}$*, labels* A*, and edges* $E = \bigcup_{i=0}^{\infty} E_i$*, which are defined recursively for all* $i \in \mathbb{N}$*:*

$$\begin{aligned} V_0 =& \bigcup_{j=1}^{n} \{B \mid B \text{ is maximally spontaneous in } P_j\} \cup \\ & \{\textstyle\bigcup_{j=1}^{n} B_j \mid \forall 1 \leq j \leq n \cdot B_j \text{ is maximally initial in } P_j\} \\ E_i =& \{(v, a, (v \setminus \{a\}) \cup \\ & \qquad \{b \mid \exists j \cdot b \in A_j^O \cup A_j^H, a \text{ triggers } b \text{ in } P_j\}) \mid v \in V_i, a \in v\} \\ V_{i+1} =& \{v' \mid v \in V_i, a \in A, (v, a, v') \in E_i, v' \notin \bigcup_{j=0}^{i} V_j\} \end{aligned}$$

We denote the subset-closure of the vertices of the approximate enabled graph (V, E) *of* C *as* $\tilde{ES}_C$*:*

$$\tilde{ES}_C = \{v' \mid v' \subset v, v \in V\}$$

The vertices of the approximate enabled graph are a superset of the vertices of the enabled graph of a distribute I/O-IMC.

Theorem 58. *Given a closed composite I/O-IMC* C*, if a set of actions controlled by* C *is a vertex in the enabled graph of* C *then a superset of that set is a vertex in the approximate enabled graph of* C*:*

$$\bar{ES}_C \subseteq \tilde{ES}_C.$$

Proof. It is enough to show that the set of vertices of EG_C is a subset of the set of vertices of $\bar{EG}_C$, i.e., any vertex v of EG_C is also a vertex of $\bar{EG}_C$. For a vertex v of EG_C we find that there is path from a vertex v_0 in V_0 of EG_C to v. We can prove Theorem 58 by induction on the length of this path using the results (8.1), (8.2), and (8.5). □

As an obvious corollary we find that the vertices of the approximate enabled graph of a closed composite I/O-IMC overapproximate the enabled sets of that I/O-IMC:

$$ES_C \subset \bar{ES}_C \subset \tilde{ES}_C.$$

To illustrate the use of approximate enabled graphs we consider again the examples discussed throughout this chapter.

Example 33. *Consider the three composite I/O-IMCs* C*,* C'*, and* C'' *from the Examples 30, 31, and 32. We can approximate the spontaneous actions, initial actions, and the triggering relation by studying the components of these composite I/O-IMCs.*

	C	C'	C''
Spontaneous sets:	$\{\{a\}, \{b\}\}$	$\{\emptyset\}$	$\{\{a\}\}$
Initial sets:	$\emptyset$	$\{\{a\}, \{b\}\}$	$\{\{d\}\}$
Trigger relation:	$\{(b, c)\}$	$\{(b, c)\}$	$\{(d, a), (d, b), (b, c)\}$

Using this information we can construct the approximate enabled graphs of the three composite I/O-IMC. They are shown in Figure 8.7. We can see that the set of actions

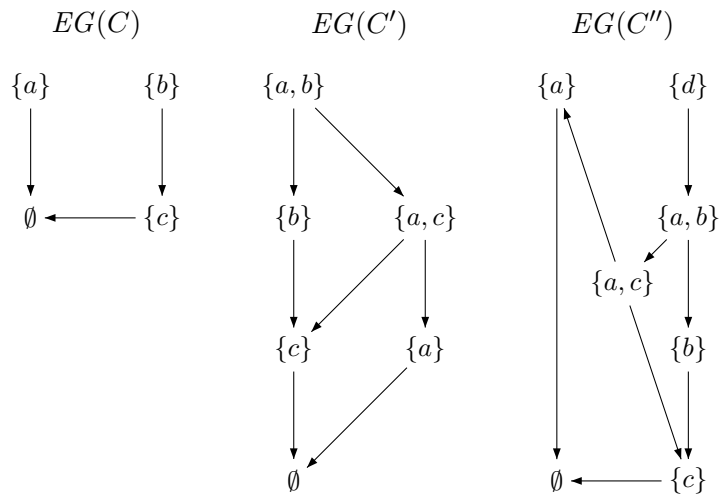


Figure 8.7: The approximate enabled graphs of composite I/O-IMCs C, C', and C''.

$\{a, b\}$ is not in the approximate enabled graph of C which conforms to the fact that $\{a, b\}$ is not an enabled set of C. On the other hand the approximate enabled graphs of C' and C'' do contain a vertex $\{a, b\}$ and indeed, $\{a, b\}$ is an enabled set of both C' and C''. However, the approximate enabled graphs of C' and C'' also contain vertices $\{a, c\}$, but $\{a, c\}$ is not an enabled set for either C' or C''.

We will now use the enabled graph to show that actions that are enabled simultaneously must have a common cause. To do this we will use the reflexive transitive closure of the triggering relation, which we call the *indirect triggering relation.*

Definition 100 (Indirect triggering relation)**.** *Given an I/O-IMC P with actions A, the* indirect triggering relation *of P is the reflexive transitive closure of the triggering relation of P. That is, an action $a \in A$ indirectly triggers an action $b \in A$ if there exists a sequence of actions $a_1, \ldots, a_n$ such that $a = a_1$, $b = a_n$ and for each $1 \leq i < n$ we have that a_i triggers a_{i+1}.*

We want to show that, if a set of actions is enabled at the same time, then these actions are indirectly triggered by a set of initial actions, or by a spontaneous set of actions. The following Lemma will help us accomplish this.

Lemma 21. *Given an I/O-IMC P with actions A and enabled graph EG_P. For an action a which is an element of a vertex v in EG_P, we have that if there is a path from a vertex v' to v in EG_P then there is an action b in v' such that b indirectly triggers a.*

Proof. We prove Lemma 21 by an induction on the length of the path from v' to v. If this path has length zero, then we have $v' = v$ and a indirectly triggers itself since the indirect triggering relation is the *reflexive* transitive closure of the triggering relation.

We then consider a path σ of length $n > 0$ and use as our induction assumption that Lemma 21 holds for paths of length smaller than n. Given that σ has length greater than zero we find a vertex v'' of EG_P, a path σ' of length $n-1$ from v' to v'', and an edge from v'' to v labelled with an action $c \in A$, such that $\sigma = \sigma' \circ (v'', c, v)$. From Definition 98 it follows that either a is in v'' or a is triggered by c. In either case we have that the vertex v'' contains an action d which indirectly triggers a, i.e., either $d = a$ or $d = c$. From our induction assumption we have that the action d must be indirectly triggered by an action b in v'. Finally, since b indirectly triggers d and d indirectly triggers a we have that b indirectly triggers a, since the indirect triggering relation is transitive. □

Now we are ready to prove that simultaneously enabled actions always have a common cause.

Theorem 59. *Given an I/O-IMC P with actions A, if actions $a_1, \ldots, a_n$ are in an enabled set of P, then there exists actions $b_1, \ldots, b_n$ that indirectly trigger a_1 through a_n respectively, such that either*

- *each of the actions $b_1, \ldots, b_n$ is initial, or*
- *P has a spontaneous set that contains all of the actions $b_1, \ldots, b_n$.*

Note that the actions $b_1, \ldots, b_n$ do not have to be distinct.

Proof. We prove Theorem 59 by considering the enabled graph of P (see Definition 98). We will assume that the actions $a_1, \ldots, a_n$ are in an enabled set of P and show that it follows that the actions $b_1, \ldots, b_n$ as above exist.

From Theorem 57 it follows that there is a vertex v of the enabled graph of P such that v is a superset of $\{a_1, \ldots, a_n\}$. Because the enabled graph of P is constructed inductively by adding edges and vertices we have that there must also exist a vertex v' in V_0 (see Definition 98) such that there is a path from v' to v in EG_P.

From Lemma 21 we know that for each action a_i, $1 \leq i \leq n$, we have that there is an action b_i in v' that indirectly triggers a_i. Finally, we see from the definition of V_0 that the vertex v' either contains only initial actions or is a spontaneous set of P. □

For composite I/O-IMCs we can approximate the indirect triggering relation in the same way as we approximated the triggering relation (see Definition 96).

Definition 101. *Given a composite I/O-IMC $C = (P_1 \| \ldots \| P_n) \backslash A_C^H$, the* approximate indirect triggering relation *is the reflexive transitive closure of the approximate triggering relation of C.*

As expected, we can now extend Theorem 59 to composite I/O-IMCs, using their indirect triggering relations.

Corollary 13. *Given a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \backslash A_C^H$, if actions $a_1, \dots, a_n$ are in an enabled set of C, then there exists actions $b_1, \dots, b_n$ that approximately indirectly trigger a_1 through a_n respectively, such that either*

- *each of the actions $b_1, \dots, b_n$ is initial in one of the components of C, or*
- *one of the components of C has a spontaneous set that contains all of the actions $b_1, \dots, b_n$.*

Note that the actions $b_1, \dots, b_n$ do not have to be distinct.

8.6 Sufficient conditions for determinism

We will now use the results we have obtained concerning the causality of interactions in I/O-IMCs to give sufficient conditions for an I/O-IMC to be weakly deterministic (see Definition 65). To be more exact, we give necessary conditions for a composite I/O-IMC to be *non-deterministic*.

Theorem 60. *Given a complete composite I/O-IMC $C = (P_1 \| \dots \| P_n) \backslash A^H$ with actions A, we have that if $\mathcal{C}$ is* not *weakly deterministic then there exists a pair of actions $a, b \in A$ such that*

1. *one of the component I/O-IMCs is* not *weakly confluent with respect to a, b,*
2. *there exist actions c, d that approximately indirectly trigger a and b, respectively, and*
3. *one of the following hold:*
   (a) *c and d are both initial actions, or*
   (b) *there exists a spontaneous set of one of the component I/O-IMCs that contains both c and d.*

Once again we note that c may be equal to d.

Proof. Theorem 60 is a refinement of Proposition 17, which states that if the composed I/O-IMC is not weakly deterministic then one of its component must be non-confluent (condition 1). However, we now take into account the fact that the non-confluence of a pair of actions is only relevant if we can actually reach a state where both actions are enabled (see Definition 65). Conditions 2 and 3 stem from Corollary 13, which gives conditions for two actions to be enabled simultaneously. □

It follows from Theorem 60 that if we cannot find a pair of actions a, b as in the theorem, then this is a sufficient condition for the composite I/O-IMC to be weakly deterministic, i.e., weakly bisimilar to an I/O-IMC with no interactive transitions. In other words, the conditions of Theorem 60 are *necessary* conditions for non-determinism. Figure 8.8 shows an example of how a non-deterministic composite I/O-IMC may fulfil

these conditions. The set $\{a\}$ is spontaneous in I/O-IMC P_1. The action a then (indirectly) triggers both actions b and c in I/O-IMCs P_2 and P_3, respectively. Finally, I/O-IMC P_4 is not weakly confluent with respect to the pair of actions b and c. In the lower right of Figure 8.8 we see that when minimised with respect to weak bisimilarity the composite I/O-IMC still contains interactive transitions.

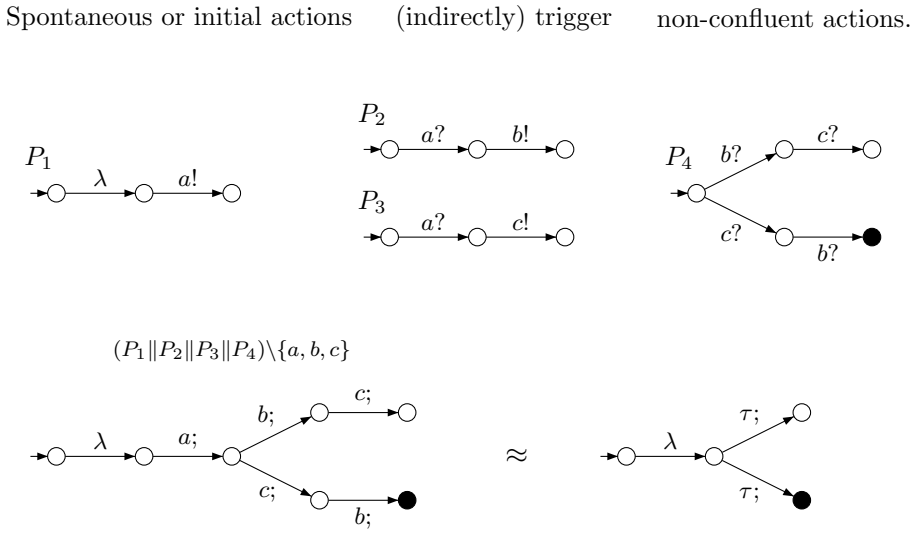


Figure 8.8: Example of a non-deterministic composite I/O-IMC $(P_1 \| P_2 \| P_3 \| P_4) \backslash \{a, b, c\}$ that satisfies the conditions of Theorem 60. All initial distributions are Dirac distributions. The colours of the states identify the state equivalence relation $=_s$.

Note that the conditions in Theorem 60 are not *sufficient* to show that an I/O-IMC is non-deterministic. Since we do not fully take into account the question of reachability we may find complete composite I/O-IMCs that fulfil the conditions of Theorem 60, but are still weakly deterministic. Figure 8.9 shows an example of such a false positive. The composite I/O-IMC in this figure satisfies the conditions in Theorem 60 in the same way as the composite I/O-IMC in Figure 8.8, but it is in fact weakly deterministic.

8.6.1 Algorithm

We will now describe an algorithm (see Algorithm 1) that verifies, for a given complete composite I/O-IMC $\mathcal{C} = (P_1 \| \ldots \| P_2) \setminus A_{\mathcal{C}}$, whether or not it satisfies the conditions in Theorem 60. Our goal is to check these conditions in polynomial time and space. Note that Theorem 60 refers to the set of all spontaneous sets of actions, which has size $O(2^{|A|})$) and would require exponential space to store. However, we are only interested in knowing whether two actions c and d are in the same spontaneous set. To

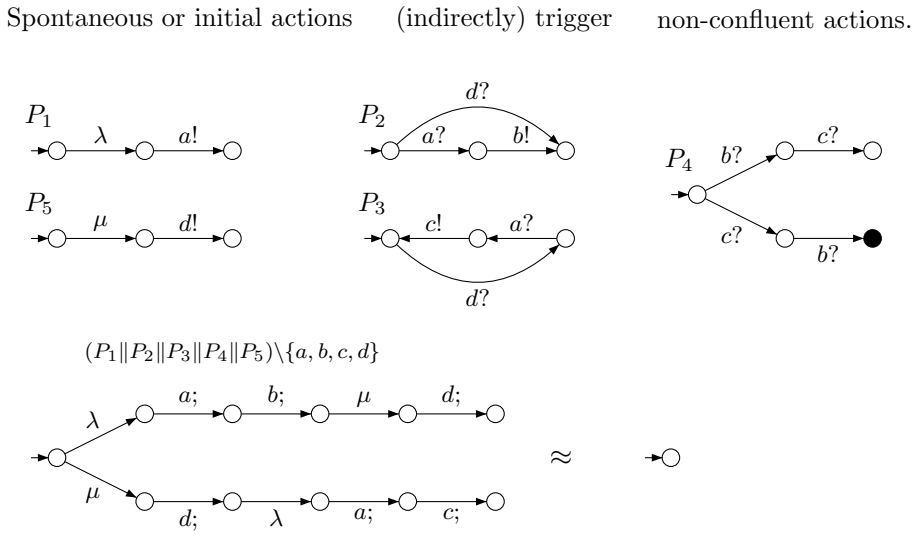


Figure 8.9: Example of a false positive for Theorem 60. All initial distributions are Dirac distributions. The colours of the states identify the state equivalence relation $=_s$.

prevent the exponential cost of computing all spontaneous sets, we therefore compute the *spontaneous relation*

$$R^{(sp)} = \{(a, b) \mid a, b \text{ are in the same spontaneous set}\}.$$

It is clear that $R^{(sp)}$ has size $O(|A|^2)$, i.e., polynomial size. We will write $R_i^{(sp)}$ for the spontaneous relation of I/O-IMC P_i. As for the initial actions, we only need to know whether an action is initial or not, so we simply compute the set of initial actions $A_i^{(init)}$ for each I/O-IMC P_i, which contains all the initial actions of that I/O-IMC and has size $O(|A|)$. Finally, we compute for each component I/O-IMC the set of all pairs of actions that are non-confluent. This set also has size $O(|A|^2)$.

Complexity Computing the spontaneous relation, initial actions, triggering relation, and non-confluent pairs of actions can be done by applying depth-first searches to the component I/O-IMCs. This requires $O(\sum_{i=1}^{n} |S_i|^2)$ time. The approximate triggering relation for $\mathcal{C}$ is simply the union of the triggering relations of its components and also has size $O(|A|^2)$. Computing the reflexive, transitive closure of this relation then has time complexity $O(|A|^3)$ [12]. Now we have all the ingredients prepared to start looking for a pair of actions a, b that satisfies the conditions of Theorem 60. We do this by considering all non-confluent pairs of actions (there are $O(|A|^2)$ such pairs) and then looking for either a pair of initial actions or a pair of spontaneous actions (again there

noend 1 Verifies whether a composite I/O-IMC $\mathcal{C} = (P_1 \| \ldots \| P_n) \backslash A_{\mathcal{C}}$. satisfies the conditions of Theorem 60. If the algorithm returns "True" then $\mathcal{C}$ may be non-deterministic, otherwise $\mathcal{C}$ is weakly deterministic.

```
1: for all 1 ≤ i ≤ n do
2:    Compute R_i^(sp), A_i^(init), and the triggering relation for P_i.
3:    Compute all pairs of non-confluent actions for P_i.
4: Compute approximate triggering relation for C.
5: Compute reflexive, transitive closure of approximate triggering relation.
6: for all non-confluent pairs of actions a, b do
7:    for all initial actions c that approximately indirectly trigger a do
8:       for all initial actions d do
9:          if d indirectly triggers b then
10:            return True
11:   for all spontaneous actions c that appr. indirectly trigger a do
12:      for all actions d in the same spontaneous set as c do
13:         if d indirectly triggers b then
14:            return True
15: return False
```

are $O(|A|^2)$ such pairs). This gives us a time-complexity of $O(|A|^4)$ for actually checking the conditions of Theorem 60 (loop 6-14). We then have an overall time complexity of

$$O(\sum_{i=1}^{n} |S_i|^2 + |A|^4)$$

and an overall space complexity of

$$O(\sum_{i=1}^{n} |S_i|^2 + |A|^2)$$

since we need to store the component I/O-IMCs and the various relations on the actions.

8.7 Time-divergence

Time-divergence in an I/O-IMC model may complicate its analysis and may also indicate a modelling issue. It is therefore useful to efficiently establish whether or not an I/O-IMC is time-divergent. Fortunately, we can use the (approximate) triggering relation of a composite I/O-IMC to show that an I/O-IMC is non-divergent.

Theorem 61. *Given a closed I/O-IMC $\mathcal{C} = (P_1 \| \ldots \| P_n) \backslash B$, if there exists no action a of $\mathcal{C}$ such that (a, a) is in the transitive closure of the (approximate) triggering relation of $\mathcal{C}$, then there is no stochastically reachable interactive cycle in $\mathcal{C}$.*

Proof. We will prove Theorem 61 by contradiction. Assume then that the I/O-IMC $\mathcal{C}$ has an interactive cycle of length $n \in \mathbb{N}$, but there is no action a of $\mathcal{C}$ such that (a, a) is in the transitive closure of the triggering relation of $\mathcal{C}$.

Let $x_1, \ldots, x_n$ and $a_1, \ldots, a_n$ be the states respectively actions on the interactive cycle. That is, the cycle is of the form

$$x_1 \xrightarrow{a_1} x_2 \ldots x_{n-1} \xrightarrow{a_{n-1}} x_n \xrightarrow{a_n} x_1.$$

Each action a_i is enabled in state x_i. For every action a_i we have that either a_i is enabled in every state on the cycle, or there exists some index j_i such that a_i is not enabled in x_{j_i} but is enabled in all states between x_{j_i} and x_i on the cycle. For the first case (a_i is enabled in every state) we have that a_i triggers itself, since there is a transition $x_i \xrightarrow{a_i} x_{i+1\%n}$ and a_i is enabled in $x_{i+1\%n}$. This obviously means that (a, a) is in the transitive closure of the triggering relation of $\mathcal{C}$ which is a contradiction.

Assume then that for each action a_i we find an index $1 \leq j_i \leq n$ as above. We have that the each action a_{j_i} triggers the action a_i since there is a transition $x_{j_i} \xrightarrow{a_{j_i}} x_{j_i+1\%n}$ and a_i is enabled in $x_{j_i+1\%n}$. It directly follows that there must be a sequence of actions from $a_1, \ldots, a_n$ that forms a triggering cycle, which is a contradiction.

Since the approximate triggering relation is a superset of the triggering relation, the fact that (a, a) is in the transitive closure of the triggering relation means that it is also in the transitive closure of the approximate triggering relation. It follows that Theorem 61 also holds for the approximate triggering relation. □

Of course, the reverse of Theorem 61 does not hold and we may find "false negatives". That is, there exist I/O-IMCs with cyclic (approximate) triggering relations, which are not time-divergent.

We can also derive a necessary condition for time-divergence from the enabled graph of $\mathcal{C}$.

Corollary 14. *Given a closed I/O-IMC $\mathcal{C} = (P_1 \| \ldots \| P_n) \backslash B$, if there are no cycles in $EG_{\mathcal{C}}$ (or $\bar{EG}_{\mathcal{C}}$) then there is no stochastically reachable interactive cycle in $\mathcal{C}$.*

Proof. We will show that whenever $\mathcal{C}$ contains a stochastically reachable interactive cycle, and (due to Theorem 61) the (approximate) triggering relation of $\mathcal{C}$ contains a "cycle", then $EG_{\mathcal{C}}$ respectively $\bar{EG}_{\mathcal{C}}$ contains a cycle.

Let $a_1, \ldots, a_n$ be the actions that constitute a cycle in the triggering relation of $\mathcal{C}$. Since the cycle is stochastically reachable we find a vertex $\{a_1\} \cup B$ in $EG_{\mathcal{C}}$, where B is a subset of the actions of $\mathcal{C}$. We then find a transition in $EG_{\mathcal{C}}$ to a vertex $\{a_2\} \cup B \cup B_1$ since a_1 triggers a_2. The set of actions B_1 consists of the other actions (possibly none) triggered by a_1. We again find a transition to a vertex $\{a_3\} \cup B \cup B_1 \cup B_2$ and so on until we reach vertex $\{a_n\} \cup B \cup B_1 \cup \ldots \cup B_{n-1}$. Since a_n triggers a_1 we now find a transition to vertex $\{a_1\} \cup B \cup \ldots$. This means that for any vertex of the form $\{a_1\} \cup B$ we can find a path in $EG_{\mathcal{C}}$ to a vertex $\{a_1\} \cup B'$ where $B' \supset B$. Since $\mathcal{C}$ has a finite number of actions it follows that we must find a cycle in $EG_{\mathcal{C}}$.

By the same reasoning, it follows that a cycle in the approximate triggering relation leads to a cycle in $\bar{EG}_{\mathcal{C}}$. □

Algorithm. To check whether a composite I/O-IMC satisfies the condition of Theorem 61 we can use an algorithm that is very similar to Algorithm 1. As in Algorithm 1 we compute the approximate triggering relation of $\mathcal{C}$ from the triggering relations of its components. We can then compute the transitive closure of this triggering relation and check whether there exists an action a such that (a, a) is in this transitive closure. This algorithm has the same space complexity as Algorithm 1 and time complexity $O(\sum_{i=1}^{n} |S_i|^2 + |A|^3)$.

8.8 Discussion

In this chapter we have developed efficient means to overapproximate the determinism and non-divergence of a composite I/O-IMC.

In Chapter 9 we will see that the theory developed in this chapter can be put to good use in practice. However, it will be very interesting to study just how "over" our over-approximations are. For instance, how often can we expect to see false positives for Theorem 60 and what aspects of a composite I/O-IMC influence the likelihood of seeing such a false positive? Our intuition is that composite I/O-IMCs with complex interactions between the component I/O-IMCs are more prone to such false positives than composite I/O-IMCs whose complexity lies mostly in their Markovian transitions.

8.8.1 Other methods to show determinism

In the literature there are several other approaches to proving that (stochastic) interacting models are deterministic. Here we briefly discuss two such approaches. Milner proposes constructing confluent LTSs by allowing only "confluent" choices between actions [35]. This approach is also possible for I/O-IMCs but is very restrictive and does not make use of the fact that, due to maximal progress, the use of non-confluent choices may not be problematic for I/O-IMCs. Bohnenkamp has shown that determinism can be ensured in stochastic process algebras by disallowing choices between interactive transitions and ensuring processes are never blocked [2]. This approach is also more restrictive than ours.

8.8.2 Determinism for networks of IMCs.

We now consider how the theory developed in this chapter applies to networks of IMCs. We first note that for composite I/O-IMCs we do not consider *renaming* of actions. We therefore consider also networks of IMCs without renaming. We conjecture that for networks of IMCs without renaming and which are deterministic with respect to synchronisation the results in this chapter also hold. For a network of IMCs with renaming, we may first determine (or overestimate) the enabled sets of the same network without renaming, before applying renaming to these enabled sets. This should lead to an overestimation of the enabled sets of the network of IMCs.

8.8.3 Practical repercussions

In this chapter we have developed an efficient way to determine whether or not a composite I/O-IMC is deterministic and non-divergent, although we may find false negatives, i.e., I/O-IMCs that are deterministic but do not satisfy the sufficient conditions presented in this chapter. However, we have been able to determine for a composite I/O-IMC consisting of many component I/O-IMCs in parallel, that it is in fact deterministic and non-divergent, using the algorithms discussed in this chapter. How does this help us in practice? One way in which this is useful is that we can in fact construct the CTMC semantics of the composed I/O-IMC *on the fly*. That is, we can compute individual transitions from the CTMC by simply applying the rules of parallel composition and resolving any "non-deterministic" choices that occur arbitrarily (since we know they will lead to bisimilar states). This allows us to apply the on-the-fly CTMC analysis techniques that we briefly discussed in Subsection 7.7.2, which may allow us to more efficiently analyse large I/O-IMCs by avoiding the generation of their state space. We will go into a bit more detail of this discussion in Chapter 9, where we consider a high-level dependability language that gives rise to complex compositions of many I/O-IMCs.

9 Arcade

In this chapter we discuss how the foundations laid in the previous chapters can be connected to an expressive modelling language. The *architectural dependability evaluation framework*, ARCADE [3], aims at providing a modular and easy-to-use architectural description language focussing on system dependability. It is designed to allow the representation of dependability features of complex systems. As such, ARCADE models describe how the components of the system may fail, how they are repaired, what the dependencies between the components are, and under what circumstances the system is considered to be operational or not. Ultimately, we are interested in computing different dependability metrics for the system, based on its behaviour.

ARCADE is equipped with a compositional semantics that maps on I/O-IMCs in such a way that each component of the dependable system is represented by one or more I/O-IMCs. The semantics of the entire system is then simply defined as the parallel composition of the I/O-IMCs representing the system components. Several of the properties established for I/O-IMCs in the preceding chapters will make it possible to establish insights about the determinism and analysability of ARCADE models.

Contribution. The design of ARCADE is rooted in joint work [13, 14, 3]. We give – for the first time – a formal syntax of ARCADE models and formalize the semantic embedding of ARCADE models into I/O-IMCs. It is known from previous work [3] that the I/O-IMC semantics of a group of ARCADE components is defined by the parallel composition of the I/O-IMC semantics of these components. However, the results from Chapter 6 allow us to extend this result: the stochastic behaviour of an ARCADE component (that is, its jump probabilities and interactive behaviour) is given by a family of interactive jump processes which can be derived from its I/O-IMC semantics and the stochastic behaviour of a group of ARCADE components is then given by the parallel composition of these jump processes. Finally, we apply the results of Chapter 8 to give

simple structural sufficient conditions for an ARCADE model to be deterministic. We will give a polynomial-time algorithm to check whether an ARCADE model satisfies these conditions.

9.1 Syntax of Arcade

The ARCADE modelling language can be used to describe complex dependable systems. In this section we will define a formal grammar for the ARCADE modelling language and we will use it to model two examples of dependable systems.

9.1.1 Formal grammar

An ARCADE model consists of a number of interacting *components*. There are five different types of components (basic components, OR-gates, AND-gates, repair units, and spare management units) and each type of component has a different set of properties, which are expressed by the following grammar.

Definition 102. *The* ARCADE *modelling language has the following grammar:*

$$
\begin{aligned}
A &::= C \mid C, A \\
C &::= \mathsf{BC} \mid \mathsf{Or} \mid \mathsf{And} \mid \mathsf{Rep} \mid \mathsf{Sp} \\
\mathsf{BC} &::= \textit{“BC”}(\mathbf{name}, \langle \textit{Omodes} \rangle, \langle \textit{Frates} \rangle, \{\textit{Fmodes}\}, \mathbf{rate}) \\
\textit{Omodes} &::= \textit{Omode}, \textit{Omodes} \mid \epsilon \\
\textit{Omode} &::= (\mathbf{name}, \mathbf{name}, \textit{signal}, \textit{signal}) \\
\textit{Frates} &::= \textit{Frate} \mid \textit{Frate}, \textit{Frates} \\
\textit{Frate} &::= \mathbf{rate} \mid \textit{“destructive”} \\
\textit{Fmodes} &::= \textit{Fmode} \mid \textit{Fmode}, \textit{Fmodes} \\
\textit{Fmode} &::= (\mathbf{name}, \mathbf{prob}) \\
\textit{signals} &::= \textit{signal} \mid \textit{signal}, \textit{signals} \\
\textit{signal} &::= \textit{Failure}(\mathbf{name}) \mid \textit{Recovery}(\mathbf{name}) \mid \textit{Failure}(\mathbf{name}, \mathbf{name}) \\
\mathsf{Or} &::= \textit{“OR”}(\mathbf{name}, \{\textit{signals}\}) \mid \textit{“OR”}(\mathbf{name}, \{\textit{signals}\}, \mathbf{name}) \\
\mathsf{And} &::= \textit{“AND”}(\mathbf{name}, \{\textit{signals}\}) \mid \textit{“AND”}(\mathbf{name}, \{\textit{signals}\}, \mathbf{name}) \\
\mathsf{Rep} &::= \textit{“REP”}(\textit{strategy}, \langle \textit{names} \rangle) \\
\textit{strategy} &::= \textit{“dedicated”} \mid \textit{“FCFS”} \mid \textit{“PP”} \mid \textit{“PNP”} \\
\mathsf{Sp} &::= \textit{“SMU”}(\mathbf{name}, \mathbf{name}, \langle \textit{names} \rangle) \\
\textit{names} &::= \mathbf{name} \mid \mathbf{name}, \textit{names}
\end{aligned}
$$

where **name** *is an identifier (such as a string),* **rate** *is a positive real, and* **prob** *is a real number between 0 and 1 (inclusive).*

We call the syntactical elements C *components* of the ARCADE model. The five different types of components are basic components (syntactical element BC), AND-gates

(syntactical element And), OR-gates (syntactical element Or), repair units (syntactical element Rep), and spare management units (syntactical element Sp). Note that the signal *Failure*(B) represents the failure of component B, whereas the signal *Recovery*(B) represents the recovery of component B. Finally, signal *Failure*(B, M) represents the failure of component B in failure mode M. We will sometimes use the shorthand $f(B)$, $u(B)$, and $f(B, M)$, to denote *Failure*(B), *Recovery*(B), and *Failure*(B, M) respectively. We will now explain the syntax of the five different types of components.

9.1.2 Basic component

A basic component models, as the name suggest, a basic component of the dependable system. A basic component is formally represented by a 5-tuple consisting of its name, operational modes, failure rates, failure modes, and repair rates.

Example 34. *As an example consider the basic component*

$$BC(\textit{Valve 1}, \emptyset, \langle 8.4 \cdot 10^{-8} \rangle, \{(\textit{stuck open}, 0.5), (\textit{stuck closed}, 0.5)\}, 0.1).$$

This basic component,

- *has* name *"Valve 1",*
- *has no* operational modes*,*
- *has a single* failure rate $8.4 \cdot 10^{-8}$*,*
- *has two* failure modes *named "stuck open" and "stuck closed" both of which occur with probability* 0.5*, and*
- *has* repair rate 0.1*.*

The name of the basic component is used to identify it. In this example, the basic component represents a valve. The failure rate of the component, describes the time until the basic component fails (in this case the basic component fails after an exponential delay with rate $8.4 \cdot 10^{-8}$*). The failure modes describe the different ways in which the valve may fail and the relative likelihoods of these failures. In our example, the valve may either become stuck open or stuck closed. Whenever the valve fails, it becomes stuck open with probability* 0.5 *and suck closed with probability* 0.5*. Finally, the repair rate tells us how fast the valve can be repaired. In this case the valve will be repaired after an exponential delay with rate* 0.1*.*

The rate at which a basic component fails may depend on its environment. For instance, in a pumping system where two pumps run in parallel, the failure of one pump will put additional strain on the remaining pump as it must pump additional water. Such outside influences are modelled using *operational modes*.

Example 35. *Consider the basic component*

$$BC(Pump\ 2, \langle (normal, degraded, u(Pump\ 1), f(Pump\ 1)) \rangle,$$
$$\langle 5.44 \cdot 10^{-6}, 10.88 \cdot 10^{-6} \rangle, \{failure, 1\}, 0.1).$$

This basic component has one operational mode

$$(normal, degraded, u(Pump\ 1), f(Pump\ 1)).$$

We can see that this basic component represent a pump which can either operate normally *or in a* degraded *manner. The two signals of the operational mode (*$u(Pump\ 1)$ *and* $f(Pump\ 1)$*) tells us under what circumstances the pump will switch from normal mode to degraded mode. In particular, when pump 1 fails, then pump 2 moves to degraded mode. Similarly, if pump 1 recovers then pump 2 moves back to normal mode. The two failure rates correspond to the two operational states (normal or degraded) of the pump. If the pump is in the "normal" state then it fails with rate* $5.44 \cdot 10^{-6}$*, if the pump is in the "degraded" state then if fails with rate* $10.88 \cdot 10^{-6}$*.*

If a basic component has multiple operational modes (such as normal/degraded as well as active/inactive), then we treat these orthogonally. That is, such a basic component would have four different combinations of operational modes: normal active, normal inactive, degraded active, and degraded inactive. We refer to these combinations of operational modes as *operational states* and we must specify a failure rate for each operational state (see Subsection 9.1.7).

If we look closely at the grammar of ARCADE we will see that there is one special failure rate, denoted by the word "destructive". Any operational state that has the "destructive" failure rate is called a *destructive operational state*, which means that whenever the basic component enters this operational state it will fail immediately. This is used to model destructive dependencies between components. For instance, it may be the case that if the power supply of a computer work station fails, this will immediately cause the work station to break down.

Note that the syntax above allows only exponential failure and repair distributions. This is done only for the sake of simplicity. In principal, ARCADE can support any distribution that is represented by a Markov chain. For more details, see previous work on dynamic fault trees [7].

9.1.3 Logical gates

Logical gates allow us to group basic components together. Each logical gate has a name and a list of *input* failures. The idea is that an OR-gate represents a complex part of the system which fails when *one* of its input failures occurs. Similarly, an AND-gate represents a complex part of the system which fails when *all* of its input failures occur. If the failure condition of a logical gate is no longer satisfied, it will recover, i.e., the corresponding subsystem is operational again. A logical gate can be embellished with an additional label (the optional third part of the tuple). In this case, the logical

gate represents an aspect of the system which we want to study. Commonly, there is one logical gate that represents complete system failure, but we could also have logical gates that represent, for instance, the failure of a subsystem. This allows us to study complex properties of the dependable system, such as the probability of system failure conditioned on the failure of a particular subsystem. In Section 9.2 we will see how these aspects are represented in the semantics of ARCADE.

9.1.4 Repair units

In ARCADE, basic components are repaired by *repair units*, which represent the real-world repair processes typically used for dependable systems(i.e., a repair unit may model a repair team tasked with replacing a faulty component). Each repair unit is responsible for the repair of a number of basic components and each repair unit has a strategy to decide which basic component to repair when multiple basic components have failed. For instance, the repair unit REP(FCFS, $\langle A, B, C\rangle$) is responsible for repairing basic components A, B, and C and will use a first-come-first-serve strategy to determine which basic component to repair first. That is, it will repair basic components in the order in which they failed. A different repair strategy is used by *prioritised preemptive* (PP) repair units. The basic components in the care of PP repair units have different priorities (given by the order in which the components are listed) and the PP repair unit will always repair the highest priority component first, preempting the repair of a lower priority component if necessary. *Prioritised non-preemptive* (PNP) repair units are similar to PP repair units, except that ongoing repairs are not preempted by the failure of a higher-priority component. We will not describe the semantics of PNP repair units in detail in this thesis as they are very similar to PP repair units. The *dedicated* repair strategy is used when the repair unit is responsible for exactly one basic component.

9.1.5 Spare management units

ARCADE also provides a mechanism to model the use of spare components using *spare management units*. A spare management unit allows several basic components to function as spares for a *primary* basic component. When the primary component fails, the first available spare is activated to take over the function of the primary. For instance, the SMU SMU($X, P, \langle A, B\rangle$) represents an SMU named X with primary component P and spares A and B. Note that several spare management units can share the same spare (e.g., the four "primary" tires of a car all share the spare tire). For the sake of simplicity, we will not discuss spare management units in detail in this chapter. For more information we refer to Boudali et al [3] and Maaß [34].

9.1.6 Other Arcade elements

Besides the ARCADE syntactical elements discussed so far, there are several other ARCADE elements as described by Boudali et al and Maaß [3, 34]. We will give a short overview of them here.

Repair units with different repair strategies. Boudali et al and Maaß discuss several more repair strategies. Prioritised non-preemptive repair units (PNP RUs) do not preempt the repair of a low priority component when a higher priority component fails. Maaß further discusses PP and PNP repair units where the priorities of the components need not form a total order [34]. When multiple components with the same priority have failed, a FCFS strategy is used to determine which of these components is repaired first.

Basic components with phase-type distributions So far, we have assumed that the failure distributions are all exponential distributions. However, ARCADE also supports the use of phase-type (PH) distributions [51, 42] to describe failure and repair distributions. A phase-type distribution consists in essence of a CTMC with a single absorbing state, where the associated distribution is the time until this absorbing state is reached. The non-absorbing states of the Markov chain are called the *phases* of the PH-distribution. An exponential distribution is a phase-type distribution with a single non-absorbing state. For a more detailed explanation of the combination of PH distributions and operational modes we refer to related work on dynamic fault trees, where a similar issue occurs [7].

Future extensions to Arcade. Since ARCADE is an extensible framework, new syntactic elements may be added to it. Such new syntactic elements must be provided with a semantics in terms of I/O-IMCs, for instance by providing a translation to MODEST descriptions [34].

9.1.7 Well-formed Arcade models

An ARCADE model is *well-formed* if it follows certain rules.

Definition 103. *An* ARCADE *model is* well-formed *if the following hold:*

1. *The model contains a finite number of syntactical elements. In particular, the number of basic components, logical gates, spare management units, and repair units is finite; each basic component has a finite number of operational modes; and each basic component has a finite number of failure modes.*
2. *The names of basic components, logical gates, and spare management units are unique.*
3. *For each basic component in the* ARCADE *model we have that*
   (a) *the number of failure rates equals two to the power of the number of operational modes, and*
   (b) *the failure probabilities sum up to exactly one.*
4. *For every signal Failure(NAME) and Recovery(NAME) appearing in the description of one the components, there exists a basic component, logical gate, or spare*

management unit with name NAME. For every signal Failure(NAME, MODE) there exists a basic component with name NAME and a failure mode MODE.

5. *For any basic component we have that its name appears at most once in the second component (the list of names) of exactly one repair unit and every name appearing in such a list corresponds to a basic component.*

6. *The second component of a spare management unit is the name of a basic component. The third component of a spare management unit forms a list of names of basic components.*

In the remainder of this chapter we consider only well-formed ARCADE models.

9.1.8 Examples of Arcade models

To show how the ARCADE modelling language works in practice, we will use it to describe two dependable systems: a pump system from a nuclear power facility and a simple replicated web service.

Pump system The pump system is responsible for pumping hot water from a nuclear reactor to a heat exchanger and pumping cool water from the heat exchanger back into the reactor. Figure 9.1 shows an overview of the pump system. For the sake of redundancy, it consists of two separate pump lines, each being capable of providing enough pumping capacity by itself. Each pump line consists of an input valve, a filter, a pump valve, a pump, and an output valve. This example is a simplified version of the pump system case study described by Boudali et al [3].

When the filter or the pump of a pump line breaks down, that pump line stops functioning. When a valve breaks down, it may either be stuck in an open position or a closed position. In the latter case, the flow of water is obstructed and the pump line stops functioning. The former case is considered to be (relatively) harmless and will not affect the functioning of the pump line. The rate at which the pumps fail depend on how much water they must pump. This means that if one of the pumps stops working, the other pump must pump more water and is more likely to fail.

Each filter or valve has a dedicated repair man that is responsible for repairing the component once it fails. There is also a single repair team for the pumps. This team can only repair one pump at the same time. This repair team uses a first-come-first-serve strategy to determine which pump to repair.

The formal ARCADE model for this example is as follows (we have left out the

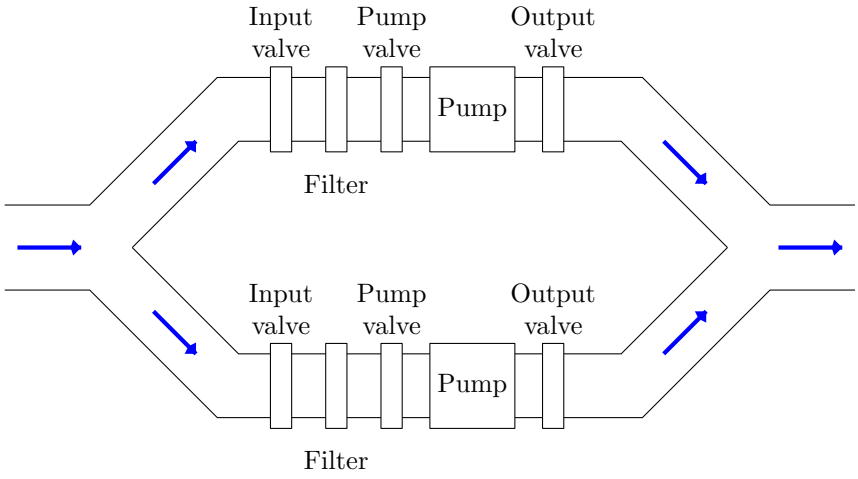


Figure 9.1: Schematic of a pump system.

remaining dedicated repair units for the sake of brevity).

$$
\begin{aligned}
&\mathrm{BC}(P1, \langle(\text{normal}, \text{degraded}, \mathit{Recovery}(P2), \mathit{Failure}(P2))\rangle,\\
&\quad \langle 5.44 \cdot 10^{-6}, 10.88 \cdot 10^{-6}\rangle, \{(\text{normal}, 1)\}, 0.1),\\
&\mathrm{BC}(P2, \langle(\text{normal}, \text{degraded}, \mathit{Recovery}(P1), \mathit{Failure}(P1))\rangle,\\
&\quad \langle 5.44 \cdot 10^{-6}, 10.88 \cdot 10^{-6}\rangle, \{(\text{normal}, 1)\}, 0.1),\\
&\mathrm{BC}(\mathit{IV1}, \emptyset, \langle 8.4 \cdot 10^{-8}\rangle, \{(\text{stuck open}, 0.5), (\text{stuck closed}, 0.5)\}, 0.1),\\
&\qquad \vdots\\
&\mathrm{BC}(\mathit{PV2}, \emptyset, \langle 8.4 \cdot 10^{-8}\rangle, \{(\text{stuck open}, 0.5), (\text{stuck closed}, 0.5)\}, 0.1),\\
&\mathrm{BC}(\mathit{F1}, \emptyset, \langle 1.14 \cdot 10^{-6}\rangle, \{(\text{normal}, 1)\}, 0.1),\\
&\mathrm{BC}(\mathit{F2}, \emptyset, \langle 1.14 \cdot 10^{-6}\rangle, \{(\text{normal}, 1)\}, 0.1),\\
&\mathrm{OR}(\mathit{PL1}, \{\mathit{Failure}(P1), \mathit{Failure}(\mathit{IV1}, \text{stuck closed}), \mathit{Failure}(\mathit{OV1}, \text{stuck closed}),\\
&\quad \mathit{Failure}(\mathit{PV1}, \text{stuck closed}), \mathit{Failure}(\mathit{F1})\}),\\
&\mathrm{OR}(\mathit{PL2}, \{\mathit{Failure}(P2), \mathit{Failure}(\mathit{IV2}, \text{stuck closed}), \mathit{Failure}(\mathit{OV2}, \text{stuck closed}),\\
&\quad \mathit{Failure}(\mathit{PV2}, \text{stuck closed}), \mathit{Failure}(\mathit{F2})\}),\\
&\mathrm{AND}(\mathit{System}, \{\mathit{Failure}(\mathit{PL1}), \mathit{Failure}(\mathit{PL2})\}, \text{no cooling}),\\
&\mathrm{REP}(\text{FCFS}, \langle P1, P2\rangle),\\
&\mathrm{REP}(\text{dedicated}, \langle \mathit{IV1}\rangle)\\
&\qquad \vdots
\end{aligned}
$$

Note, that in the original presentation of this example, the pump components had Erlang-distributed failure and repair times instead of exponentially distributed ones [3].

For the sake of simplicity we have taken exponential distributions here. As mentioned in Subsection 9.1.6, the syntax of ARCADE can easily be extended to allow more general distributions.

Replicated web service Our second example is a replicated web service which consists of two types of components: HTTP servers, which handle incoming request from users accessing a website and database (DB) servers that store data for the service. The HTTP servers may access the database servers to store or retrieve information. There are two power supplies, one for the web servers and one for the database servers. Figure 9.2 shows a schematic of the replicated web service.

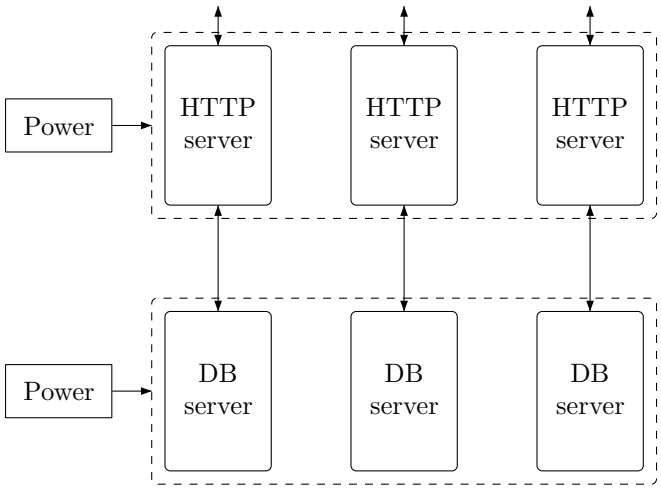


Figure 9.2: Schematic of a replicated web service. Arrows denote data-flow and dashed boxes denote the two power supply groups.

Our distributed web service comprises eight basic events. Three HTTP servers, three database servers, and two power supplies. HTTP servers and database servers can be either *powered* or *unpowered* depending on whether their power supply is operational or not. The powered/unpowered switch is modelled as an *operational mode*. The unpowered operational state is a *destructive* operational state, which means that when a server (abruptly) loses power it will immediately become inoperable and will require maintenance (i.e., it must be repaired) to become operational again.

Each of the power supplies has a dedicated repair unit. Further, there is a single repair unit for all of the HTTP servers and a single repair unit for all of the database servers. Both of these repair units use a first-come-first-serve strategy to determine which server is repaired first.

The HTTP servers and database servers make up three pairs of servers, each capable of providing the web servers. Each such pair is grouped together by an OR-gate, since if either the HTTP server or the database server breaks down, this pair will be unable to provide the web service. The web service itself is considered to be down if neither of the

three pairs of HTTP server and database server is operational. The web service is then modelled as an AND-gate, with the three OR-gates representing the pairs of servers as inputs.

The ARCADE model for this example if given below.

$$
\begin{aligned}
&\mathrm{BC}(\mathit{P1}, \emptyset, \langle 0.001 \rangle, \{\text{normal}, 1\}, 2),\\
&\mathrm{BC}(\mathit{P2}, \emptyset, \langle 0.001 \rangle, \{\text{normal}, 1\}, 2),\\
&\mathrm{BC}(\mathit{H1}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P1}), \mathit{Failure}(\mathit{P1}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{BC}(\mathit{H2}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P1}), \mathit{Failure}(\mathit{P1}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{BC}(\mathit{H3}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P1}), \mathit{Failure}(\mathit{P1}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{BC}(\mathit{D1}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P2}), \mathit{Failure}(\mathit{P2}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{BC}(\mathit{D2}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P2}), \mathit{Failure}(\mathit{P2}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{BC}(\mathit{D3}, \langle(\text{powered}, \text{unpowered}, \mathit{Recovery}(\mathit{P2}), \mathit{Failure}(\mathit{P2}))\rangle,\\
&\quad \langle 0.03, \text{destructive}\rangle, \{(\text{normal}, 1)\}, 5),\\
&\mathrm{OR}(\mathit{S1}, \{\mathit{Failure}(H1), \mathit{Failure}(\mathit{D1})\}),\\
&\mathrm{OR}(\mathit{S2}, \{\mathit{Failure}(H2), \mathit{Failure}(\mathit{D2})\}),\\
&\mathrm{OR}(\mathit{S3}, \{\mathit{Failure}(H3), \mathit{Failure}(\mathit{D3})\}),\\
&\mathrm{AND}(\mathit{System}, \{\mathit{Failure}(S1), \mathit{Failure}(\mathit{S2}), \mathit{Failure}(\mathit{S3})\}, \text{service unavailable}),\\
&\mathrm{REP}(\text{FCFS}, \langle \mathit{H1}, \mathit{H2}, \mathit{H3}\rangle),\\
&\mathrm{REP}(\text{FCFS}, \langle \mathit{D1}, \mathit{D2}, \mathit{D3}\rangle),\\
&\mathrm{REP}(\text{dedicated}, \langle \mathit{P1}\rangle),\\
&\mathrm{REP}(\text{dedicated}, \langle \mathit{P2}\rangle)
\end{aligned}
$$

9.2 Operational behaviour of Arcade

This section describes the semantics of ARCADE. When designing the language, we identified four main building blocks with which we can, in a modular fashion, construct a system dependability model: (1) a Basic Component (BC), (2) a Repair Unit (RU), (3) a Logical Gate (LG), and (4) a Spare Management Unit (SMU). These building blocks interact with each other by sending and receiving input/output events. The semantics of these building blocks and their interactions is based on the I/O-IMC framework. For each syntactical ARCADE elements, we will find one (or more) I/O-IMCs that describe the semantics of the syntactical element. In this section we will describe the semantics of basic components, repair units, and logical gates. We will denote the I/O-IMC semantics

of an ARCADE syntactical element X as

$$[\![X]\!].$$

As mentioned, we will not cover the semantics of spare management units and we cover the semantics of repair units only for FCFS, PP, and dedicated repair strategies. For a full description of the ARCADE semantics in terms of I/O-IMCs we refer to Boudali et al and Maaß [3, 34].

9.2.1 Basic component

The basic component building block represents a physical/logical system component that has a distinct operational and failure behaviour. Before introducing the semantics of basic components in formal terms, we first discuss some essential concepts. A basic component has two, almost orthogonal aspects: (1) its failure model, i.e., the different ways in which the component may fail, and (2) its operational modes, i.e., the behaviour of the component when it is operational. Throughout this subsection we will consider a basic component with $n \in \mathbb{N}$ operational modes and $m > 0 \in \mathbb{N}$ failure modes, given by

$$\mathrm{BC}(B, \langle o_1, \ldots, o_n \rangle, \langle \lambda_1, \ldots, \lambda_{2^n} \rangle, \{(F_1, p_1), \ldots, (F_m, p_m)\}, \mu). \tag{9.1}$$

When constructing the state space of the I/O-IMC semantics of a basic component, we will consider the failure model of the BC and the operational model of the BC separately. The states of the failure model describe whether the basic component is operational or not. However, we also need states to indicate that the basic component is about to become (in)operational. Finally, we have to distinguish between the different failure modes of the basic component.

Definition 104. *Given a BC as in* (9.1)*, its* failure states *are the states in the set*

$$\mathcal{FS} = \{UP, FAILING, DOWN, REC\} \cup \{FAILING(F_i) \mid 0 < i \leq m\}.$$

The state $FAILING(F_i)$ indicates that the BC is failing in the failure mode F_i. We use the state $FAILING$ to indicate a failure due to a destructive dependency. The state REC indicates that the BC is recovering from a failure.

Almost orthogonal to the failure model of a BC is its *operational model*, which describes the changes to the operational modes of the BC. The set of *operational states* is the enumeration of all different combinations of operational modes.

Definition 105. *Given a BC as in* (9.1)*, its* operational states *are the states in the set*

$$\mathcal{OS} = \{0, 1\}^n.$$

We use a simply binary encoding to construct the operational states. For a BC with 3 operational modes active/inactive, powered/unpowered, and normal/degraded, the operational state $(0, 1, 0)$ corresponds to the BC being active, unpowered, and normal. The operational state of a BC is controlled by the signals associated with the operational modes. Each such signal corresponds to an I/O-IMC action.

Definition 106. *Given a signal S as defined in Definition 102, we find a corresponding action $a(S)$ as follows:*

$$a(S) = \begin{cases} f_B, & S = \mathit{Failure}(B), \\ u_B, & S = \mathit{Recovery}(B), \\ f_B^{(M)}, & S = \mathit{Failure}(B, M). \end{cases}$$

We can then define the mode-switching actions as follows.

Definition 107. *Given a BC as in* 9.1 *with operational modes $o_1 = (M_1, M_1', S_1, S_1')$, $o_2 = (M_2, M_2', S_2, S_2')$, ..., its* mode switching actions *are the actions $a(S_1), a(S_2), \ldots$ and $a(S_1'), a(S_2'), \ldots$. For convenience we will denote the mode-switching actions using the following scheme: $a_1 = a(S_1), a_2 = a(S_2), \ldots$ and $b_1 = a(S_1'), b_2 = a(S_2'), \ldots$.*

We distinguish between two different types of operational state: normal operational states represent an operational state of the basic component in which it will fail spontaneously after some stochastic delay. On the other hand, *destructive* operational modes represent operational states in which the basic component fails immediately.

Definition 108. *Given a BC as in* 9.1*, we order its operational states $\mathcal{OS}$ lexicographically: $s_0 = \langle 0, \ldots, 0, 0\rangle$, $s_1 = \langle 0, \ldots, 0, 1\rangle$, $s_2 = \langle 0, \ldots, 1, 0\rangle$, etc. We then find for each operational state s_i its failure rate λ_i. The set of* destructive *operational states $\mathcal{OS}_d$ is defined as:*

$$\mathcal{OS}_d = \{s_i \mid \lambda_i = \text{“destructive”}, 0 \leq i < 2^n\}.$$

If an operational state is not destructive it is normal *and we then find the set of normal operational states:*

$$\mathcal{OS}_m = \mathcal{OS} \setminus \mathcal{OS}_d.$$

We are now ready to define the semantics of a basic component in terms of its underlying I/O-IMC.

Definition 109. *Given a basic component with $n \in \mathbb{N}$ operational modes and $m > 0 \in \mathbb{N}$ failure modes, given by*

$$BC(B, \langle o_1, \ldots, o_n\rangle, \langle \lambda_1, \ldots, \lambda_{2^n}\rangle, \{(M_1, p_1), \ldots, (M_m, p_m)\}, \mu),$$

with $o_i = (m_i, m_i', S_i, S_i')$ for all $0 \leq i < n$, its I/O-IMC semantics is the I/O-IMC

$$\begin{aligned} P &= [\![BC(B, \langle o_1, \ldots, o_n\rangle, \langle \lambda_1, \ldots, \lambda_{2^n}\rangle, \{(M_1, p_1), \ldots, (M_m, p_m)\}, \mu)]\!] \\ &= \langle S, A, R^I, R^M, \alpha\rangle, \end{aligned}$$

with

- *state space*

$$\begin{aligned} S &= \mathcal{FS} \times \mathcal{OS} \\ &= \{REC, UP, DOWN\} \cup \{(FAILING_j) \mid 1 \leq j \leq m\} \times \{0, 1\}^n, \end{aligned}$$

- *actions*

$$A^I = \{a_i, b_i \mid 1 \leq i \leq n\} \cup \{r_B\}$$
$$A^O = \{f_B^{(M_j)} \mid 1 \leq j \leq m\} \cup \{f_B, u_B\}$$
$$A^H = \{\tau_B\},$$

  where $a_i = a(S_i)$ *and* $b_i = a(S_i')$ *for all* $1 \leq i \leq n$,

- *Markovian transitions*

$$\{((UP, \mathbf{m}), p_i\lambda_{\mathbf{m}}, (FAILING(M_i), \mathbf{m})) \mid \mathbf{m} \in \mathcal{OS}_m,\ i \in [1, m]\} \quad (9.2)$$

- *Interactive transitions*

$$\{(UP, \mathbf{m}), \tau_B, (FAILING, \mathbf{m}) \mid \mathbf{m} \in \mathcal{OS}_d\} \quad (9.3)$$
$$\cup\{((FAILING(M_i), \mathbf{m}), f_B^{(M_i)}, (DOWN, \mathbf{m})) \mid \mathbf{m} \in \mathcal{OS},\ i \in [1, m]\} \quad (9.4)$$
$$\cup\{((FAILING, \mathbf{m}), f_B, (DOWN, \mathbf{m})) \mid \mathbf{m} \in \mathcal{OS}\} \quad (9.5)$$
$$\cup\{((DOWN, \mathbf{m}), r_B, (REC, \mathbf{m})) \mid \mathbf{m} \in \mathcal{OS}\} \quad (9.6)$$
$$\cup\{((REC, \mathbf{m}), u_B, (UP, \mathbf{m})) \mid \mathbf{m} \in \mathcal{OS}\} \quad (9.7)$$
$$\cup\{((x, \mathbf{m}), r_B, (x, \mathbf{m}) \mid \mathbf{m} \in \mathcal{OS},\ x \in \mathcal{FS} \setminus \{DOWN\}\} \quad (9.8)$$
$$\cup\{((x, (e_1, ..., e_j, ..., e_n)), a_j, (x, (e_1, ..., 0, ..., e_n))) \mid x \in \mathcal{FS},\ j \in [1, n]\} \quad (9.9)$$
$$\cup\{((x, (e_1, ..., e_j, ..., e_n)), b_j, (x, (e_1, ..., 1, ..., e_n))) \mid x \in \mathcal{FS},\ j \in [1, n]\} \quad (9.10)$$

- *and an initial distribution which assigns probability one to the state* $(UP, (0, \ldots, 0))$ *and probability zero to all other states.*

Example 36. *Figure 9.3 shows the I/O-IMC*

$$[\![BC(B, \langle (m, m', Recovery(B'), Failure(B')\rangle, \langle \lambda, destructive\rangle, \{(p, X), (1-p, Y)\}, \mu)]\!].$$

The I/O-IMC in Figure 9.3 has the following states:

$$x_0 = (UP, \langle 0\rangle) \qquad x_6 = (UP, \langle 1\rangle)$$
$$x_1 = (FAILING(X), \langle 0\rangle) \qquad x_7 = (FAILING(X), \langle 1\rangle)$$
$$x_2 = (FAILING(Y), \langle 0\rangle) \qquad x_8 = (FAILING(Y), \langle 1\rangle)$$
$$x_3 = (FAILING, \langle 0\rangle) \qquad x_9 = (FAILING, \langle 1\rangle)$$
$$x_4 = (DOWN, \langle 0\rangle) \qquad x_{10} = (DOWN, \langle 1\rangle)$$
$$x_5 = (REC, \langle 0\rangle) \qquad x_{11} = (REC, \langle 1\rangle).$$

For clarity we have left out the following mode-switching transitions:

$$(x_1, f_{B'}, x_7), (x_7, u_{B'}, x_1),$$
$$(x_2, f_{B'}, x_8), (x_8, u_{B'}, x_2),$$
$$(x_3, f_{B'}, x_9), (x_9, u_{B'}, x_3),$$
$$(x_5, f_{B'}, x_{11}), (x_{11}, u_{B'}, x_5).$$

Note that the recovery transitions $(x_4, r_B?, x_5)$ *and* $(x_{10}, r_B?, x_{11})$ *are interactive and not Markovian. Although the repair of a basic component will always take place after some delay, this delay is not represented in the semantics of the basic component itself. Instead, this repair-delay is represented in the I/O-IMC semantics of the various repair units (e.g., see. Section 9.2.3).*

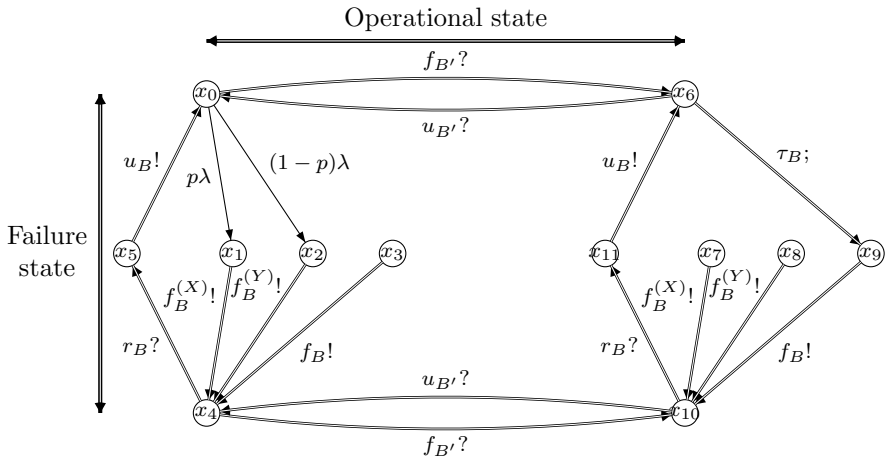


Figure 9.3: Basic component with two failure modes and two operational states. Several mode-switching transitions have been left out for the sake of readability.

9.2.2 Logical gates.

In this subsection we will consider the I/O-IMC semantics of *logical gates*. Recall that a logical gate allows us to group certain basic components together as subsystems. A logical gate describes the relationship between failure (and recovery) of the subsystem and failure (and recovery) of its components. Here we will only discuss AND- and OR-gates, but these gates can be used to construct more complex logical operators such as the *voting* gate [34].

The AND-gate represents a subsystem that fails only when all of its components have broken down. The OR-gate on the other hand represents a subsystem that fails when just one of its components has broken down. The components that make up a subsystem represented by a logical gate are called the *inputs* of that logical gate.

Logical gates can also be used to describe fundamental properties of the dependable system. Most commonly, one of the logical gates will describe whether the complete system has failed or not. For instance, we can use an AND-gate to express the fact that (in the pump system example from Section 9.1) the system fails when the first pump train *and* the second pump train are down. We will model such system properties by using the state equivalence relation $=_s$. In this way, two I/O-IMC states that represent system states that differ in one of the system properties (e.g., one state where the

complete system has failed and one state where the complete system is still operational) will not be equivalent according to $=_s$ and thus will never be weakly bisimilar.

We will first consider an AND-gate X with n inputs which models a property ϕ:

$$\mathrm{AND}(X, \{\mathit{Failure}(B_1), \mathit{Failure}(B_2), \ldots, \mathit{Failure}(B_n)\}, \phi). \tag{9.11}$$

We say that $B_1, \ldots, B_n$ are the *inputs* of the AND-gate. The I/O-IMC that describes the operational behaviour of an AND-gate keeps track of the state of all its inputs and then fires failure and recovery actions when necessary.

Definition 110. *The I/O-IMC semantics of the AND gate X given in (9.11) is the I/O-IMC*

$$\begin{aligned} P &= [\![\mathit{AND}(X, \{\mathit{Failure}(B_1), \mathit{Failure}(B_2), \ldots, \mathit{Failure}(B_n)\}, \phi)]\!] \\ &= \langle S, A, R^I, R^M, \alpha \rangle, \end{aligned}$$

with

- *state space,*
$$S = \{UP, DOWN\}^n \times \{UP, DOWN\}$$
- *actions,*
$$\begin{cases} A^I = \{f_{B_1}, \ldots, f_{B_n}\} \cup \{u_{B_1}, \ldots, u_{B_n}\}, \\ A^O = \{f_X, u_X\}, \\ A^H = \emptyset, \end{cases}$$
- *interactive transitions,*
$$\{((\mathbf{x}, y), f_{B_i}, (\mathbf{x}', y)) \mid x_i' = DOWN, x_j' = x_j \text{ for all } j \neq i\} \tag{9.12}$$
$$\cup\{((\mathbf{x}, y), u_{B_i}, (\mathbf{x}', y)) \mid x_i' = UP, x_j' = x_j \text{ for all } j \neq i\} \tag{9.13}$$
$$\cup\{((\mathbf{x}, UP), f_X, (\mathbf{x}, DOWN)) \mid \mathbf{x} = DOWN^n\} \tag{9.14}$$
$$\cup\{((\mathbf{x}, DOWN), u_X, (\mathbf{x}, UP)) \mid \mathbf{x} \neq DOWN^n\} \tag{9.15}$$

  where $1 \leq i, j \leq n$ and $y \in \{UP, DOWN\}$,
- *no Markovian transitions, and*
- *an initial distribution which assigns probability one to state (UP^n, UP) and zero to all other states.*

Finally, the following states will be labelled ϕ:

$$\{(\mathbf{v}, DOWN) \mid \mathbf{v} \in \{UP, DOWN\}^n\}.$$

Each state of the I/O-IMC describing an AND-gate with n inputs consists of two components: a vector of length n describing the operational status (UP or $DOWN$) of its n inputs. The second component describes the operational status (again, UP or $DOWN$) of the AND-gate itself. Whenever one of the input-actions (f_{B_i} or u_{B_i}) occur, the operational status of the i-th input is changed accordingly (9.12 and 9.13). The output action f_X (9.14) is fired whenever the AND-gate is up but all of its inputs are down. Similarly, the output action u_X (9.15)) is fired whenever the AND-gate is down but at least one of its inputs is up.

The I/O-IMC describing an OR-gate with n inputs is very similar to the I/O-IMC in Definition 110 except that the condition that must always hold is: the OR-gate is down if and only if one or more of its inputs are down. We then find the same I/O-IMC semantics as in Definition 110 except that the sets of transitions 9.14 and 9.15 are replaced by

$$\{((\mathbf{x}, UP), f_X, (\mathbf{x}, DOWN)) \mid \mathbf{x} \neq UP^n\} \tag{9.16}$$

$$\cup\{((\mathbf{x}, DOWN), u_X, (\mathbf{x}, UP)) \mid \mathbf{x} = UP^n\} \tag{9.17}$$

Example 37. *Figure 9.4 shows the I/O-IMC*

$$[\![AND(X, \{Failure(B_1), Failure(B_2)\}, \phi)]\!].$$

Note that several of its states are labelled with ϕ.

9.2.3 Dedicated repair units

In a dependable system it is usually the case that components can be repaired to ensure that the system can recover from small-scale failures. In ARCADE, repair units model the various ways in which components of a system can be repaired. Each repair unit is responsible for repairing one or more basic components and each basic component has at most one repair unit associated with it.

A repair unit that is responsible for repairing only one basic component is called a *dedicated repair unit*. We consider a dedicated repair unit that is responsible for the basic component B. Let $F_1, \ldots, F_m$ be the $m \in \mathbb{N}$ failure modes of B and let μ be the repair rate of B.

Definition 111. *The operational behaviour of a dedicated repair unit*

$$REP(\text{“dedicated”}, \{B\})$$

is the I/O-IMC

$$\begin{aligned} P &= [\![REP(\text{“dedicated”}, \{B\})]\!] \\ &= \langle S, A, R^I, R^M, \alpha \rangle, \end{aligned}$$

with

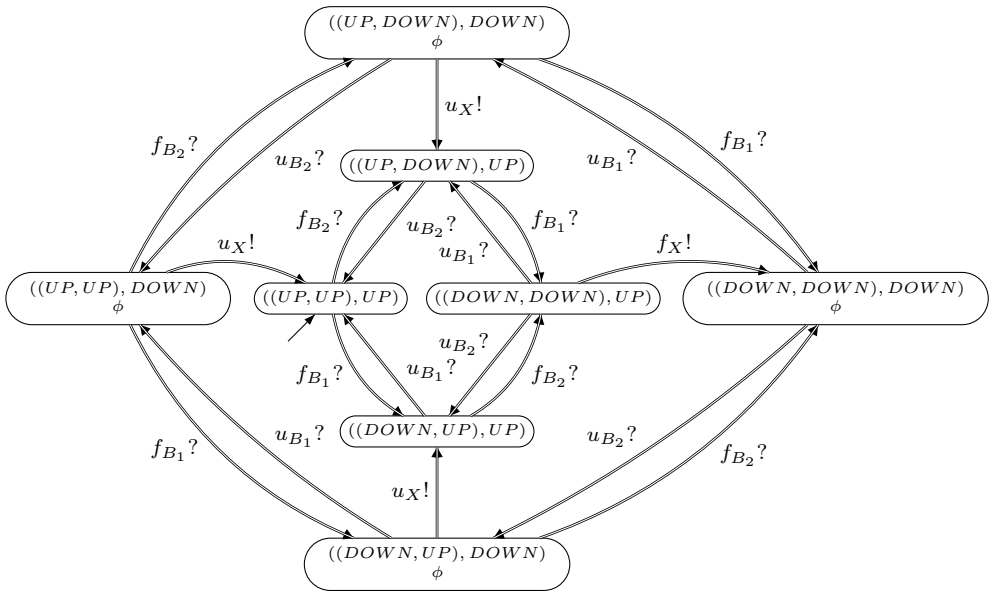


Figure 9.4: Example of an AND-gate with inputs B_1 and B_2, which models the system property ϕ. That is, ϕ holds when both B_1 *and* B_2 are "down" or to be more precise, ϕ holds when the AND-gate is in its "down" state.

- *state space,*
$$S = \{UP, DOWN, DONE\}$$
- *actions,*
$$\begin{cases} A^I = \{f_B\} \cup \{f_B^{(F_i)} \mid 1 \leq i \leq m\}, \\ A^O = \{r_B\}, \\ A^H = \emptyset, \end{cases}$$
- *interactive transitions,*
$$\begin{aligned} &\{(x, f_B, DOWN) \mid x \in S\} \\ \cup &\{(x, f_B^{(F_i)}, DOWN) \mid x \in S, 1 \leq i \leq m\} \\ \cup &\{(DONE, r_B, UP)\} \end{aligned}$$
- *Markovian transitions,*
$$\{(DOWN, \mu_B, DONE)\},$$
where μ_B is the repair-rate for basic component B, and
- *an initial distribution which assigns probability one to state UP and zero to all other states.*

Initially, the repair unit is idle. A failure of the basic component associated with this repair unit is signalled by input-actions f_B and $f_B^{(F_i)}$. When the repair unit receives such a signal it will start *repairing*. The repair is finished after an exponentially distributed delay. This delay is characterised by the *repair rate* μ_B. The greater this repair rate is the faster the repair unit will repair the basic component. Finally, the completion of the repair is signalled by the output action $r_B!$.

The transitions of the I/O-IMC

$$[\![\text{REP}(\text{“dedicated”}, \langle B \rangle)]\!],$$

where B has one failure mode F can be seen in Figure 9.5. The states UP and $DOWN$ describe the state of the basic component that the repair unit is responsible for. The state $DONE$ signifies that the RU has just finished repairing the BC.

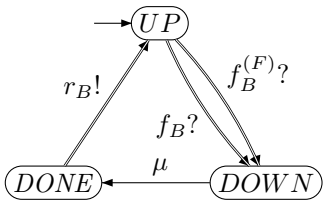


Figure 9.5: Example of the I/O-IMC semantics of a dedicated repair unit for a BC B_1 with one failure mode F and repair rate μ.

9.2.4 Preemptive prioritised repair unit.

When a repair unit is responsible for more than one basic component things get more interesting. Whenever more than one of these basic component is down, the repair unit will have to decide which one to repair first as we assume the repair unit can undertake only one repair at the same time. There are many different strategies to decide which component to repair first. We will discuss two of them here.

Definition 112. *The operational behaviour of a preemptive prioritised repair unit responsible for n basic components,*

$$RU(\text{“}PP\text{”}, \langle B_1, \ldots, B_n \rangle),$$

where for each $1 \leq i \leq n$ the basic component B_i has $m_i \in \mathbb{N}$ failure modes $F_1, \ldots, F_{m_i}$ and repair rate μ_{B_i}, is the I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$, with

- *state space,*

$$S = \{UP, DOWN\}^n \times (\{DONE_i \mid 1 \leq i \leq n\} \cup \{BUSY\}).$$

- *actions,*

$$A^I = \{f_{B_i} \mid 1 \leq i \leq n\} \cup \{f_{B_i}^{(F_j)} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}$$
$$A^O = \{r_{B_i} \mid 1 \leq i \leq n\}$$
$$A^H = \emptyset$$

- *interactive transitions,*

$$\begin{aligned}&\{((\mathbf{x}, y), f_{B_i}, (\mathbf{x}', y)) \mid x_i' = DOWN, x_j' = x_j, j \neq i\}\\ \cup&\{((\mathbf{x}, y), f_{B_i}^{(F_k)}, (\mathbf{x}', y)) \mid x_i' = DOWN, x_j' = x_j, j \neq i, 1 \leq k \leq m_i\}\\ \cup&\{((\mathbf{x}, DONE_i), r_{B_i}, (\mathbf{x}', BUSY)) \mid x_i' = UP\}\end{aligned}$$

  where $1 \leq i \leq n$ *and* $y \in \{DONE_i \mid 1 \leq i \leq n\} \cup \{BUSY\}$,

- *Markovian transitions,*

$$\{((\mathbf{x}, BUSY), \mu_{B_i}, (\mathbf{x}, DONE_i)) \mid x_i = DOWN, j < i \text{ implies } x_j = UP\},$$

  where $1 < i \leq n$, *and*

- *an initial distribution which assigns probability one to state* $(UP^n, BUSY)$ *and zero to all other states.*

The basic components that are associated with a *prioritised repair unit* are ordered based on their priority. Components with a higher priority will be repaired before components with a lower priority. The order is determined by the list of components of the repair unit. The component that is listed first has the highest priority while the component that is listed last has the lowest priority. It remains to decide what happens when a high-priority component fails while the repair unit is repairing a low-priority component. *Preemptive* prioritised (PP) repair units will preempt the repair of the low-priority component and switch to repairing the high-priority components. Non-preemptive prioritised repair units are discussed by Maaß [34].

Example 38. *Figure 9.6 shows an example of the I/O-IMC semantics of a PP repair unit which is responsible for two basic components* B_1 *and* B_2 *with respective repair rates* μ_1 *and* μ_2. *For the sake of simplicity we show only one failure action per basic component (*f_1 *respectively* f_2*). We can see that the repair unit "keeps track" of which of its BCs are operational (*UP*) or not (*$DOWN$*). The repair unit always repairs the BC with the highest priority first. We can clearly see this in the state* $((DOWN, DOWN), BUSY)$, *which has an outgoing* μ_1 *transition, meaning that the RU is repairing basic component* B_1.

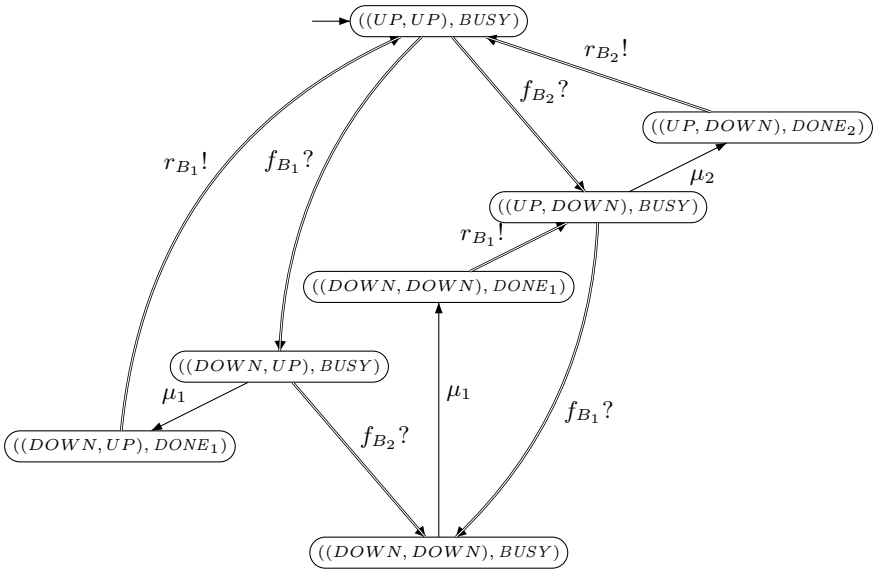


Figure 9.6: Example of the I/O-IMC semantics of a PP repair unit for two BCs b_1 and b_2 with repair rates μ_1 respectively μ_2. The BC b_1 has a higher priority than the BC b_2. For the sake of simplicity we have left out several unreachable states, the failure transitions emanating from *DONE* states, and failure transitions for the failure modes of the basic components.

9.2.5 First-come-first-serve repair units

When a repair unit gives equal priority to all the basic components that it is responsible for, then an obvious repair strategy is first-come-first-serve (FCFS). Using this strategy, the components are repaired in the order in which they fail. For a FCFS repair unit, the order in which the components are listed in is syntax has no special meaning.

To define the operational behaviour of a FCFS repair unit, we will use the following notation; $LIST(\langle B_1, \ldots, B_n \rangle)$ denotes the set of all lists containing only elements from $\langle B_1, \ldots, B_n \rangle$, none of which may appear more than once in the list. Given a list l and a component $B_i, 1 \leq i \leq n$, not present in l, we use the following notation; $l : B_i$ denotes the list obtained by appending B_i to l. For a non-empty list l, $head(l)$ denotes the first element of l, and $tail(l)$ denotes the list obtained by removing the first element of l.

Definition 113. *The operational behaviour of a first-come-first-serve repair unit responsible for n basic components,*

$$RU(\text{“}FCFS\text{”}, \langle B_1, \ldots, B_n \rangle)$$

where for each $1 \leq i \leq n$ the basic component B_i has $m_i \in \mathbb{N}$ failure modes $F_1, \ldots, F_{m_i}$ and repair rate μ_{B_i} is the I/O-IMC $P = \langle S, A, R^I, R^M, \alpha \rangle$, with

- *state space,*

$$S = LIST([1, n]) \times (\{DONE_i \mid 1 \leq i \leq n\} \cup \{BUSY\}),$$

- *actions,*

$$\begin{aligned} A^I &= \{f_{B_i} \mid 1 \leq i \leq n\} \cup \{f_{B_i}^{(F_j)} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\} \\ A^O &= \{r_{B_i} \mid 1 \leq i \leq n\} \\ A^H &= \emptyset \end{aligned}$$

- *interactive transitions,*

$$\begin{aligned} &\{((l,y), f_{B_i}, (l : B_i, y)) \mid B_i \notin l\} \\ \cup &\{((l,y), f_{B_i}^{(F_j)}, (l : B_i, y)) \mid B_i \notin l, 1 \leq j \leq m_i\} \\ \cup &\{((l,y), f_{B_i}, (l, y)) \mid B_i \in l\} \\ \cup &\{((l,y), f_{B_i}^{(F_j)}, (l, y)) \mid B_i \in l, 1 \leq j \leq m_i\} \\ \cup &\{((l, DONE_i), r_{B_i}, (tail(l), BUSY)) \mid l \neq \epsilon, head(l) = B_i\}, \end{aligned}$$

  where $1 \leq i \leq n$ *and* $y \in \{DONE_i \mid 1 \leq i \leq n\} \cup \{BUSY\}$

- *Markovian transitions,*

$$\{((l, BUSY), \mu_{B_i}, (l, DONE_i)) \mid l \neq \epsilon, head(l) = i, 1 \leq i \leq n\},$$

  and

- *an initial distribution which assigns probability one to state* $(\epsilon, BUSY)$*, where* ϵ *is the empty list and zero to all other states.*

The state space for FCFS repair units is very similar to the state space of PP repair units with the difference that the FCFS repair unit records in which *order* the basic components fail. The state of the basic components is then given by a *list* of failed components. Any basic component not in the list is operational. The order of the components in the list describes the order in which they have failed. The first basic component in the list has failed earliest.

Example 39. *Figure 9.7 shows an example of the I/O-IMC semantics of a FCFS repair unit responsible for two BCs:*

$$RU(\text{“}FCFS\text{”}, \langle B_1, B_2 \rangle).$$

The BCs have repair rates μ_{B_1} *respectively* μ_{B_2} *and to simplify the figure we show only a single failure action (*f_{B_1} *respectively* f_{B_2}*) for both of the basic components. It is somewhat similar to the I/O-IMC semantics of a PP repair unit (see Figure 9.6), except that the FCFS repair unit keeps track not only of the state of its BCs but also of the order in which they fail.*

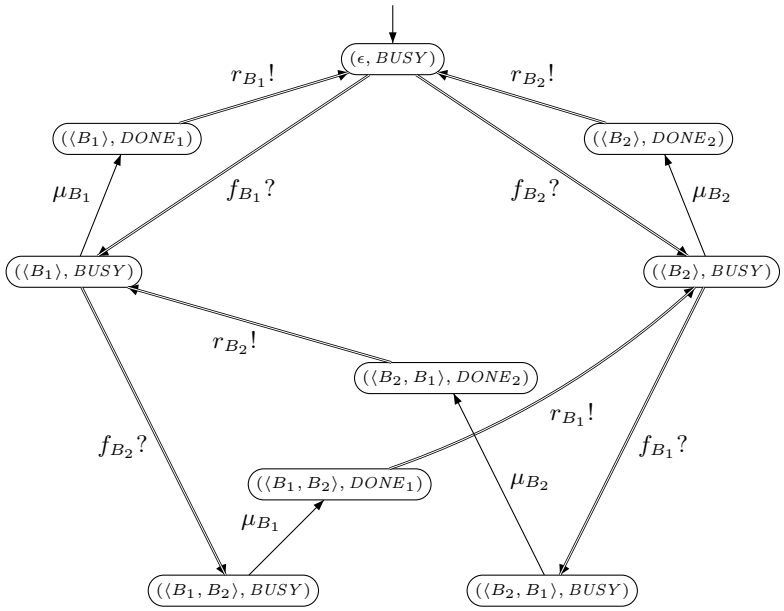


Figure 9.7: Example of the I/O-IMC semantics of a FCFS repair unit for two BCs B_1 and B_2 with repair rates μ_1 respectively μ_2. For the sake of simplicity we have left out the failure transitions emanating from *DONE* states and the failure transitions corresponding to the different failure modes of the basic components.

9.2.6 Operational semantics of an Arcade model

The semantics of a complete ARCADE model is simply the parallel composition of the semantics (I/O-IMCs) of its components. That is, for an ARCADE model consisting of components $X_1, X_2, \ldots, X_n$ for some $n \in \mathbb{N}$ we have

$$[\![X_1, X_2, \ldots, X_n]\!] = [\![X_1]\!] \| [\![X_2]\!] \| \ldots \| [\![X_n]\!].$$

Recall from Section 5.3 that parallel composition is only defined for *compatible* I/O-IMCs, that is, I/O-IMCs whose output actions are not shared and whose internal actions are unique. Fortunately, the I/O-IMC semantics of the components of a well-formed ARCADE model are indeed pair-wise compatible.

Theorem 62. *Given a well-formed* ARCADE *model without spare management units, the I/O-IMC semantics of its components are pair-wise compatible.*

Proof. Given Definitions 107, 110, 111, 112, and 113, the actions of I/O-IMCs which represent the behaviour of ARCADE components are:

- Actions of the form f_X, where X is the name of a basic component or logical gate; an action f_X is never an internal action and is an output for the basic component or logical gate with name X. The fact that names are unique for well-formed ARCADE models means these outputs are never shared.

- Actions of the form $f_X^{(M)}$, where X is the name of a basic component and M is a failure mode of X; as for actions of the form f_X, we have that an action $f_X^{(M)}$ is never an internal action and is an output for the basic component with name X. The uniqueness of names again guarantees that these outputs are never shared.

- Actions of the form u_X, where X is the name of a basic component or logical gate; as for actions of the form f_X we have that these actions never appear as internal and are output actions only for the unique component whose name is X.

- Actions of the form r_X, where X is the name of a basic component. Such actions never appear as internal actions and are output actions for any repair unit, which is responsible for repairing the basic component X. Since the name X may appear in the list of basic components of at most one repair unit, we have that the output r_X cannot be shared by more than one repair unit.

- Actions of the form τ_X, where X is the name of a basic component. These actions appear only as internal action of the I/O-IMC representing the basic component named X. Again we have that the uniqueness of names guarantees that the action τ_X is unique.

□

We are now ready to define the operational semantics of an ARCADE model.

Definition 114. *Given an* ARCADE *model* $X_1, X_2, | \ldots, X_n$ *for some* $n \in \mathbb{N}$*, its operational behaviour is the I/O-IMC obtained by composing in parallel the I/O-IMCs representing the operational semantics of its components:*

$$[\![X_1, X_2, \ldots, X_n]\!] = ([\![X_1]\!] \| [\![X_2]\!] \| \ldots \| [\![X_n]\!]) \setminus A^O,$$

where A^O *is the set of all output actions of the I/O-IMC* $[\![X_1]\!] \| [\![X_2]\!] \| \ldots \| [\![X_n]\!]$*. Recall that the states of the I/O-IMCs representing logical gates may be labelled to indicate system properties. The state-labels of a state of the I/O-IMC representing the entire* ARCADE *model are obtained by taking the set of all labels of its constituent states.*

In Chapter 7 we have seen that *closed* I/O-IMCs (I/O-IMCs without input actions) correspond to either Markov chains or Markov decision processes and can be analysed using standard solution techniques. It is then interesting to know which ARCADE models correspond to closed I/O-IMCs. We will call such ARCADE models *closed* ARCADE models.

Definition 115. *An* ARCADE *model is* closed *if the name of each basic component in the model appears* exactly *once in the list of names of a repair unit of the* ARCADE *model.*

Note that we can allow closed ARCADE models where certain basic components are never repaired by adding a type of repair unit with a "do-not-repair" strategy to

ARCADE. The I/O-IMC semantics of such a repair unit would simply be an I/O-IMC which has the expected repair actions as output actions, but has no transition, i.e., it does nothing. As is to be expected, the operational behaviour of a closed ARCADE model is a closed I/O-IMC.

Theorem 63. *The I/O-IMC semantics of a well-formed closed* ARCADE *model without spare management units is a* closed *I/O-IMC.*

Proof. From Section 5.3 we know that a composite I/O-IMC is closed if every *input* action of one of its constituent I/O-IMCs also appears as an output action for one of its other constituent I/O-IMCs. We will consider each input action appearing in the Definitions 107, 110, 111, 112, and 113 to find the I/O-IMCs that use these actions as outputs.

- A basic component named X has input action r_X. Any repair unit that has X in its list of repaired components has r_X as an output action. Our assumption that X must appear in exactly one such lists ensures that r_X is an output of exactly one I/O-IMC in the parallel composition. The basic component X may also have input actions of the form $f_Y, f_Y^{(M)}, u_Y$ corresponding to its mode-switching signals, but the fact that each mode-switching signal must correspond to a basic component or logical gate in the ARCADE model, means that for these actions we also find I/O-IMCs in the parallel composition which use the action as output.
- A logical gate has input actions $f_{Y_1}, \ldots, f_{Y_n}$ and $u_{Y_1}, \ldots, u_{Y_n}$ where $Y_1, \ldots, Y_n$ are its inputs. The fact that the inputs of the logical gate correspond to basic components and other logical gates of the ARCADE model means these actions also appear as output actions in the parallel composition.
- A repair units has input actions $f_{Y_1}, \ldots, f_{Y_n}$, and $f_{Y_i}^{M_1}, \ldots, f_{Y_i}^{M_{m_i}}$ where $Y_1, \ldots, Y_n$ are the basic components the repair unit is responsible for and $M_1, \ldots, M_{m_i}$ are the failure modes of the i-th basic component. The fact that these basic components must be part of the ARCADE model means that the corresponding failure actions must appear as output actions of the I/O-IMCs representing these basic components.

□

We conjecture that Theorems 62 and 63 also hold in the presence of spare management units. Note that, since all the output actions of the I/O-IMC representing an ARCADE model are hidden; this I/O-IMC is in fact *complete*, that is, it has only internal actions.

Example 40. *Figure 9.8 shows a schematic of the* ARCADE *model*

$$BC(B_1, \emptyset, \langle \lambda_1 \rangle, \{(M, 1)\}, \mu_1),$$
$$BC(B_2, \emptyset, \langle \lambda_2 \rangle, \{(M, 1)\}, \mu_2),$$
$$AG(L, \{B_1, B_2\}, \textit{“system failure”}),$$
$$RU(\textit{“dedicated”}, \langle B_1 \rangle), RU(\textit{“dedicated”}, \langle B_2 \rangle)$$

with two BCs (BC 1 and BC 2), which are repaired by dedicated repair units RU 1 respectively RU 2. The system is considered to be down when both BCs are down. This is modelled by an AND gate which has BCs 1 and 2 as inputs. The BCs each have a single failure mode with failure rates λ_1 respectively λ_2 and repair rates μ_1 respectively μ_2.

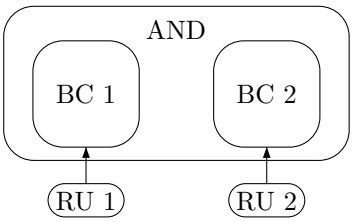


Figure 9.8: Schematic representation of an ARCADE model with two basic components, an AND-gate which describes system failure and two dedicated repair units.

The I/O-IMC semantics of the components of this ARCADE *model can be found in Figures 9.9 (top, basic components), 9.4 (AND gate), and 9.5 (dedicated repair units). The I/O-IMC semantics of the* ARCADE *model itself is then simply the parallel composition of these component I/O-IMCs where all actions are hidden. Recall that certain states of the AND-gate I/O-IMC will be labelled to indicate that the system itself is down. Figure 9.9 shows this parallel composition on the left-hand side. On the right-hand side we can see that, when minimised with respect to weak bisimulation, the semantics of our example is a simple four-state CTMC.*

9.3 Triple compositionality

We have seen in the previous section that the compositional syntax of ARCADE has a compositional semantics in terms of I/O-IMCs. However, we know from Chapter 6 that I/O-IMCs themselves also have a compositional semantics in terms of interactive jump processes. We call this property *triple compositionality*: ARCADE has a compositional syntax, a compositional semantics in terms of I/O-IMCs (which we will refer to as its *operational* semantics) and a compositional semantics in terms of interactive jump processes (which we will refer to as its *stochastic* semantics). We will show by an example that this has some nice consequences for understanding and analysing ARCADE models.

Example 41. *Consider a well-formed* ARCADE *model with a basic component B that*

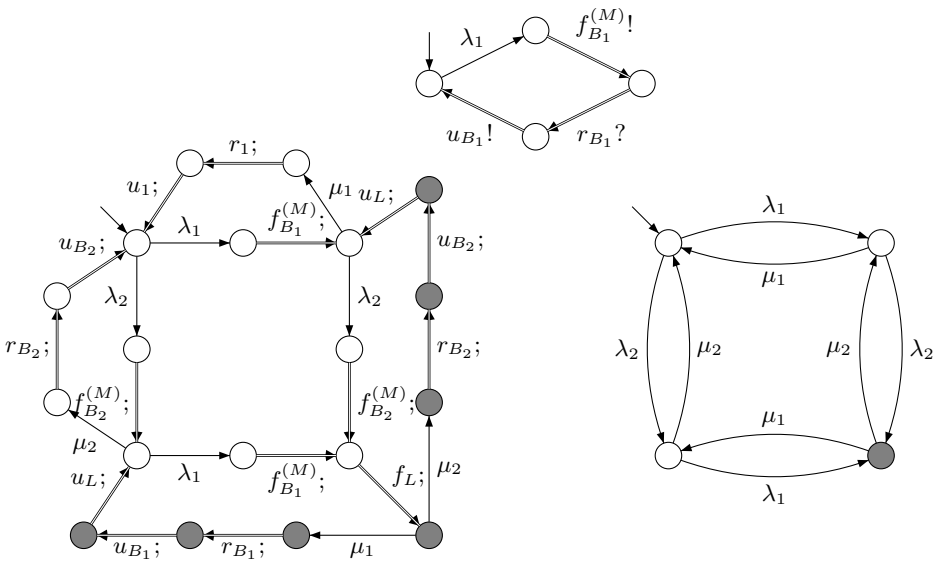


Figure 9.9: I/O-IMC semantics of a basic component (top), the ARCADE model (left), and the minimised I/O-IMC semantics of the ARCADEmodel (right). States labelled "system failure" are coloured grey. Stochastically unreachable states have been omitted.

has a dedicated repair unit

$$BC(B, \emptyset, \langle\lambda\rangle, \{(M, 1)\}, \mu),$$
$$RU(\text{“}dedicated\text{”}, \langle B\rangle),$$
$$\vdots$$

We have purposely omitted the rest of the ARCADE *model, because it turns out we can still make statements about the stochastic properties of the basic component and its repair unit regardless of how the remainder of the* ARCADE *model is chosen.*

Figure 9.10 shows the I/O-IMC semantics of the basic component B and its dedicated repair unit as well as the parallel composition of these two I/O-IMCs, which of course corresponds to the semantics of both syntactical elements put together.

The interesting thing about the combined semantics of basic component B and its repair unit is that is is a closed, deterministic I/O-IMC. This means that its stochastic behaviour is a single interactive jump process, which will emit a failure signal after an exponential delay with rate λ followed by a recovery signal after another exponential delay with rate μ, and so on. This holds no matter how the rest of the ARCADE *model is chosen.*

In fact, we can analyse the transient distribution of the interactive jump process for the I/O-IMC using standard CTMC analysis techniques and, because our modular semantics of I/O-IMCs is sound with respect to composition, these results will also apply to the complete ARCADE *model. As a very simple example we find that the probability that our basic component fails at least once within t time-units is $1 - e^{-\lambda t}$. Now recall*

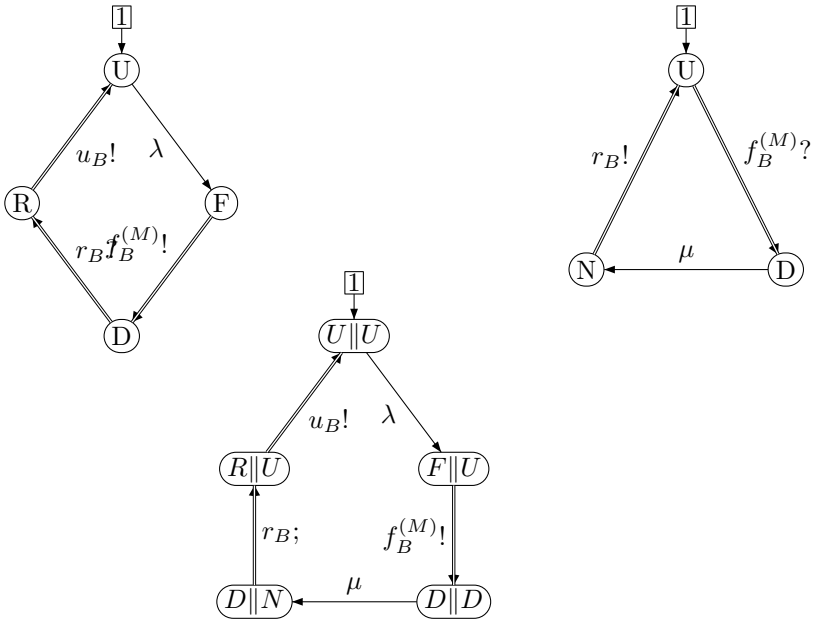


Figure 9.10: I/O-IMC semantics of a basic component, its repair unit, and their parallel composition. The names of the states have been abbreviated.

that Proposition 22 told us that transient probability bounds for a composite I/O-IMC are always tighter than the same bounds for its components. In our example, we know that the probability of at least one failure of B is exactly $1 - e^{-\lambda t}$*, which means that the same must hold for any I/O-IMC obtained by composing the I/O-IMC in Figure 9.10 with any other I/O-IMC. Additionally the same will hold for any* Arcade *model which contains the basic component and its dedicated repair unit.*

The observation in Example 41, that a basic component without failure modes and its dedicated repair unit are independent of the rest of the Arcade model they are part of and behave like a simple CTMC, is of course not surprising and should intuitively be the case assuming that the semantics of basic component and repair unit are defined sensibly. However, the work done in Chapters 6 and 7 for the first time gives us the tools to indeed prove that this intuition is correct. Furthermore, we can apply similar reasoning to other, more complicated sub-sets of Arcade models which are (mostly) independent of the rest of the system.

9.4 Causality

In the previous section, we have seen that the operational behaviour of a well-formed, closed Arcade model corresponds to a complete I/O-IMC. We have seen in Chapter 7 that such an I/O-IMC corresponds to either a CTMDP or, if the I/O-IMC is determin-

istic, a CTMC.

In Section 9.5, we will give sufficient conditions on an ARCADE model that ensure that its operational behaviour is a deterministic I/O-IMC. To prepare for this, we will in this section study the causal relationships between the different actions of I/O-IMCs representing ARCADE components. Our investigation will be based on the results from Chapter 8. Recall that Theorem 65 states that there are three necessary ingredients for any case of non-determinism in a composite I/O-IMC. We must have a spontaneous set of actions (or a combination of initial sets), these spontaneous actions must indirectly trigger two (possibly different) actions, and these two triggered actions must be non-confluent. In order to determine whether the I/O-IMC semantics of an ARCADE model satisfies these conditions, we must then study the spontaneous sets, initial sets, triggering relation, and confluence properties of the semantics of its components.

9.4.1 Basic components

We will now investigate the causal relationships between the actions of an I/O-IMC representing a basic component:

$$[\![\mathrm{BC}(B, \{o_1, \ldots, o_n\}, \langle \lambda_1, \ldots, \lambda_{2^n} \rangle, \{(F_1, p_1), \ldots, (F_m, p_m)\}, \mu)]\!]$$

with mode-switching actions $a_1, \ldots, a_n$ respectively $b_1, \ldots, b_n$ as given by Definition 107.

Spontaneous actions. All the failure actions $f_B^{(F_i)}$, where $1 \leq i \leq n$, are spontaneous (as singleton sets), since the transitions (9.2) enable these actions. Otherwise, no actions or sets of actions are spontaneous, since the transitions (9.2) enable no other actions and are the only Markovian transitions.

Initial actions. Recall that initially, with probability one, the basic component is in state $(UP, \mathbf{m})$, where $\mathbf{m}$ is the initial operational state. In case the operational state $\mathbf{m}$ is a regular operational state we have that no actions are enabled in the initial state and there are no initial actions. If, on the other hand, the initial operational state is destructive, then the action τ_B will be enabled initially and $\{\tau_B\}$ will be the only initial set of actions.

Triggering relation. It is easy to see that the internal action τ_B triggers the action f_B (through transitions (9.3)) and that the repair action r_B triggers the recovery action u_B (through transitions (9.6)). It remains to determine which actions, if any, trigger the internal action τ_B. We know that τ_B is enabled in the states $(UP, \mathbf{m})$, whenever the operational state $\mathbf{m}$ is destructive. The transitions to these states are transitions of types (9.9) and (9.10) which change the operational state to $\mathbf{m}$ or the transition (9.7) when the component is already in operational state $\mathbf{m}$. Of course, an action only triggers the action τ_B if there exists a transition labelled with that action from a state where τ_B is not enabled to a state where it is. This means that the actions a_i, b_j which may

change the operational state from non-destructive to destructive all trigger τ_B. We then find the following triggering relation.

$$\{(\tau_B, f_B), (r_B, u_B)\} \cup \{(a_i, \tau_B) \mid i \in D\} \cup \{(b_i, \tau_B) \mid i \in U\}.$$

where the set D and U are the sets of operational modes that might cause a destructive failure when they are switched by mode-switching action a respectively b, i.e.,

$$\begin{aligned} D = \{i \mid \exists \mathbf{m} \in \{0,1\}^{n-1} \cdot \langle m_1, \ldots, m_{i-1}, 0, m_i, \ldots m_{n-1}\rangle \notin \mathcal{OS}_d, \\ \langle m_1, \ldots, m_{i-1}, 1, m_i, \ldots, m_{n-1}\rangle \in \mathcal{OS}_d\} \\ U = \{i \mid \exists \mathbf{m} \in \{0,1\}^{n-1} \cdot \langle m_1, \ldots, m_{i-1}, 1, m_i, \ldots m_{n-1}\rangle \notin \mathcal{OS}_d, \\ \langle m_1, \ldots, m_{i-1}, 0, m_i, \ldots, m_{n-1}\rangle \in \mathcal{OS}_d\}. \end{aligned}$$

Note that it is possible that both a_i and b_i trigger τ_B for some $1 \leq i \leq n$. Similarly, it is possible that neither triggers τ_B.

Confluence. We now consider whether the various pairs of actions of a basic component are weakly confluent or not. First, note that there are no states where two immediate actions are enabled at the same time. The immediate consequence is that the I/O-IMC is weakly confluent with respect to all pairs of immediate (output or internal) actions.

For the operational switching actions $a_i, b_i, 1 \leq i \leq n$, it is easy to see that the I/O-IMC semantics of a basic component is weakly confluent with respect to any pair of such actions, since the transitions (9.9) and (9.10) commute in all cases. We further note that at most one failure transition (that is a transition from (9.2), (9.3), (9.4), (9.5), (9.6), (9.7), or (9.8)) can be enabled at the same time.

It then remains to check whether or not the failure transitions commute with the operational switching transitions. This does not matter for transitions (9.2), since they are Markovian. For transitions (9.4), (9.5), (9.6), and (9.7) we have that these commute with the operational switching transitions (9.9) and (9.10), since they are not influenced by (and do not influence) the operational state of the basic component. That is, these two sets of transitions are completely orthogonal.

However, this is not the case for the transitions (9.3). The enabledness of these transitions depends on whether he basic component is in a destructive operational state or not. Since transitions (9.9) and (9.10) may change the operational state of the basic component they may not commute with transitions (9.3). This means that the I/O-IMC semantics of a basic component might *not* be weakly confluent with respect to a pair of actions a_i and τ_B or a pair of actions b_i and τ_B, for some $1 \leq i \leq n$. Note that this non-confluence only occurs for such actions a_i and b_i that may turn a destructive operational state into a non-destructive operational state.

Figure 9.11 illustrates these cases of non-confluence. The mode-switching action a leads from a destructive operational state to a normal one, while the action b leads from a normal operational state to a destructive one (note that it could also be the case that the a transition leads to a destructive operational state while the b transition leads to

a normal one). In state x we can observe the non-confluence between actions a and τ_B. After following the transition labelled a, the action τ_B is no longer enabled, which means these two actions are non-confluent.

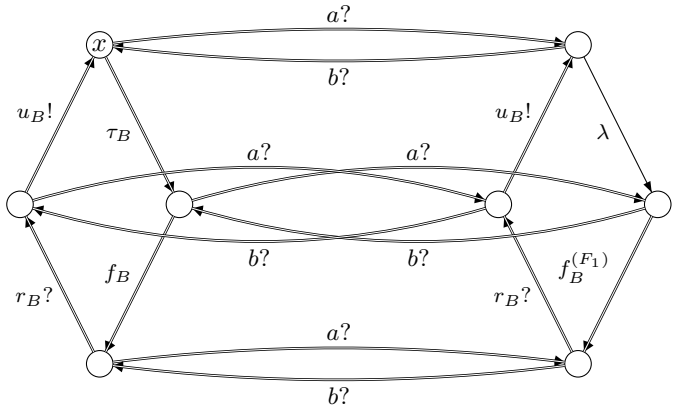


Figure 9.11: Illustration of non-confluence between the action τ_B and the mode-switching action a when switching between destructive and non-destructive operational states. Not all transitions are shown.

Table 9.1 summarises the causality and confluence properties of the I/O-IMC semantics of a BC.

Spontaneous	Initial	Triggering	Non-confluence
$\{f_B^{(F_k)}\}$	$[\{\tau_B\}]$	a_j triggers τ_B b_i triggers τ_B r_B triggers u_B τ_B triggers f_B	a_i & τ_B, b_j & τ_B

Table 9.1: Causality and confluence results for a basic component with m failure modes and n operational modes. We have $1 \leq k \leq m$ and $1 \leq i, j \leq n$. The set of initial sets is empty if the initial operational state is non-destructive. Index i is such that mode-switching action a_i leads from a destructive operational state to a normal operational and index j is such that action a_j does the reverse.

9.4.2 Logical gates

We now study the causal relationships between the actions of the I/O-IMC representing the semantics of an AND-gate with n inputs

$$[\![\mathrm{AG}(L, \{\mathit{Failure}(B_1, M_1), \ldots, \mathit{Failure}(B_n, M_n)\})]\!].$$

It is easy to see that the I/O-IMCs representing AND- and OR-gates have no initial or spontaneous actions. As for triggering, we see (for both I/O-IMCs) that each action $f_{B_i}^{M_i}$ triggers the action f_L and each action u_{B_i} triggers the action u_L. A similar result holds for logical gates with inputs of the form $\mathit{Failure}(B_i)$.

With respect to confluence, it is clear that the failure and recovery actions of the subcomponents of the logical gates are pair-wise confluent. The situation is different for the failure and recovery actions of the logical gates themselves. For instance, consider the I/O-IMC semantics of an AND-gate L when all basic components are down, but the logical gate is still up (i.e., state $(DOWN^n, UP)$). In this state we see that there is non-confluence between f_L and the actions u_{B_i} for $1 \leq i \leq n$. This is caused by the fact that once an action u_{B_i} occurs, the action f_L will no longer be enabled. Similarly we find non-confluence between the actions u_L and $f_{B_i}^{(M_i)}$ for all $1 \leq i \leq n$. However, we will see in Section 9.5 that these non-confluent actions can never be enabled at the same time. Table 9.2 describes the causality and confluence properties of the I/O-IMC semantics of a logical gate. Whether or not the logical gate represents a system property (and as a consequence has some of its states labelled) does not change its confluence properties. The causality and confluence properties of an OR-gate are the same as those of an AND-gate.

Spontaneous	Initial	Triggering	Non-confluence
$\emptyset$	$\emptyset$	$f_{B_i}^{(M_i)}$ triggers f_L	f_L & u_{B_i}
		f_{B_i} triggers f_L	u_L & f_{B_i}
		u_{B_i} triggers u_L	u_L & $f_{B_i}^{(M_i)}$

Table 9.2: Causality and confluence results for a logical gate with n basic events $B_1, \ldots, B_n$ as inputs. We have $1 \leq i \leq n$.

9.4.3 Dedicated repair units

For the I/O-IMC

$$[\![\mathrm{RU}(\text{“dedicated”}, \langle B \rangle)]\!],$$

which represents the operational behaviour of a dedicated repair unit we find that the output action r_B is spontaneous and is not triggered by any other action. The I/O-IMC is *not* confluent with respect to pairs of actions f_B and r_B as well as $f_B^{(M_j)}$ and r_B for failure modes M_j of B, since such pairs of actions do not commute in state DONE.

9.4.4 Preemptive prioritised repair units

For the I/O-IMC representing the operational behaviour of a PP repair unit

$$[\![\mathrm{RU}(\text{“PP”}, \langle B_1, \ldots, B_n \rangle)]\!],$$

where for each $1 \leq i \leq n$ we have that the basic component has $m_i \in \mathbb{N}$ failure modes $M_1, \ldots, M_{m_i}$, we have that its causal relationships are very simple. First of all, we find the spontaneous sets

$$\{\{r_{B_i}\} \mid 1 \leq i \leq n\}$$

and an empty triggering relation.

Concerning confluence, we easily see that each pair of failure actions f_{B_i}, f_{B_j} or $f_{B_i}^{(M_k)}, f_{B_j}$ and so forth is confluent. On the other hand, each repair action r_{B_i} is *not* confluent with its failure counterparts f_{B_i}, $f_{B_i}^{(M_1)}$, etc. This is due to the fact that a failure action is "ignored" when the corresponding component is already down, but recorded when it is up and, when a repair action happens the state of the component changes from down (do not record failure) to up (do record failure). Finally, each repair action r_{B_i} is confluent with all other failure actions f_{B_j}, $f_{B_j}^{(M_1)}, \ldots$ where $i \neq j$. We have summarised the causality and confluence properties of PP repair units in Table 9.3.

Spontaneous	Initial	Triggering	Non-confluence
$\{r_{B_i}\}$	$\emptyset$	$\emptyset$	r_{B_i} & f_{B_i}

Table 9.3: Causality and confluence results for a PP repair units with n basic components $B_1, \ldots, B_n$. We have $1 \leq i \leq n$. For simplicity we specify only one failure action per basic component, but all other failure actions of the same basic component have identical causality and confluence properties.

9.4.5 First-come-first-serve repair units

As for PP repair units, there are no initial actions, the sets $\{r_{B_i}\}$ are maximally spontaneous, and the triggering relation is empty.

However, the situation with regard to confluence is now somewhat different. Particularly, pairs of failure actions f_{B_i}, f_{B_j} or $f_{B_i}^{(F_k)}, f_{B_j}$ and so forth are *not* confluent for the I/O-IMC semantics of a FCFS repair units. This is due to the fact that adding index i to a list l and then adding index j to the same list yields a different result than first adding j and then i. This was not the case for the PP repair unit where the non-operational basic components were represented by a set rather than a list. The other observations with respect to confluence remain the same as for the PP repair unit. Table 9.4 summarises the causality and confluence properties of I/O-IMCs representing FCFS repair units.

9.5 Deterministic Arcade models

We will now discuss sufficient conditions under which ARCADE models have a deterministic semantics. That is, if we construct the I/O-IMC semantics of such a deterministic ARCADE model consisting of basic components, logical gates, and dedicated-, PP-, and

Spontaneous	Initial	Triggering	Non-confluence
$\{r_{B_i}\}$	$\emptyset$	$\emptyset$	r_{B_i} & f_{B_i} f_{B_i} & f_{B_j}

Table 9.4: Causality and confluence results for a FCFS repair unit with n basis components $b_1, \ldots, b_n$. We have $1 \leq i, j \leq n$ and $i \neq j$. For simplicity we specify only one failure action per basic component, but all other failure actions of the same basic component have identical causality and confluence properties.

FCFS-repair units, every interactive scheduler will yield the same transient distributions over the system properties. The conditions will be based on the one hand on the sufficient conditions for determinism of distributed I/O-IMCs described in Subsection 8.6 and on the other hand on an assumption we are willing to make about the nature of the ARCADE models.

9.5.1 Destruction by failure assumption

We make the following assumption: every mode-switching action that can cause a basic component to switch from a normal operational mode to a destructive operational mode is a failure action of a basic event or logical gate and every mode switching action that can cause a switch from a destructive operational mode to a normal operational mode is a recovery action.

This assumption ensures that the immediate destruction of a basic component can only be caused by the failure of another basic component or a subsystem (logical gate). We will refer to this assumption as the *destruction by failure* assumption. Whenever we have two basic components B_i, B_j, such that the failure of B_j (signalled, for instance, by action $f_{B_j}^{(M)}$) can cause an immediate failure in B_i we say that B_i *has a destructive dependency on* B_j. A basic component can also destructively depend on a logical gate.

Example 42. *Consider an* ARCADE *model with 3 BCs: pump P, control valve V, and power supply S. The pump has two operational modes normal/stressed and powered/unpowered. The first operational mode is controlled by the control valve. If the control valve is operational the pump is in normal mode, otherwise it is stressed. The powered/unpowered operational mode is controlled by the power supply. If the power supply is operational, the pump is powered otherwise it is unpowered. Each basic component has a single failure mode M. Figure 9.12 shows the operational state changes of the pump in two cases.*

Consider the case that the pump immediately fails when it is both stressed and unpowered and all other operational states are normal (left-hand side of Figure 9.12). We can then see that the actions $f_V^{(M)}$ and $f_S^{(M)}$ can cause a switch from a normal operational mode to a destructive operational mode. This ARCADE *model satisfies the destruction by failure assumption. The pump has a destructive dependency on both the valve and the power source.*

Consider also the case that the pump fails immediately when it is operating normally and becomes unpowered (right-hand side of Figure 9.12), but not in any other case (i.e., the other operational modes are normal). Now we see that the actions u_V and $f_S^{(M)}$ can switch the pump from a normal operational mode to a destructive operational mode. In this case, the ARCADE *model does not satisfy the destruction by failure assumption.*

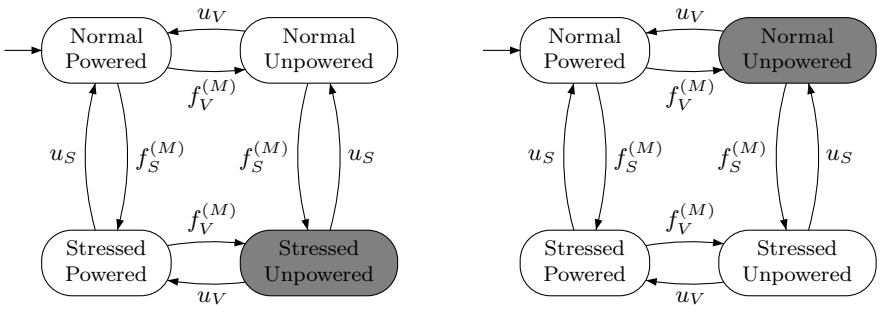


Figure 9.12: Overview of the operational states of two basic components. Destructive operational states are coloured grey. The left case satisfies the failure by destruction assumption while the right case does not.

We will leave it to the reader to show that, under the destruction by failure assumption, any basic component with an initial destructive operational state, has only destructive operational states. Recall that initially all basic component and logical gates are operational.

9.5.2 Spontaneous and initial actions

To accomplish our goal of establishing conditions that guarantee that an ARCADE model is deterministic, we first investigate which actions are *spontaneous* and *initial* (see Definitions 93 and 94) for an ARCADE model. Throughout this section we will use the causality and confluence properties for the individual ARCADE elements as summarised in Tables 9.1, 9.2, 9.3, and 9.4. We will consider a well-formed ARCADE model with basic events $B_1, \ldots, B_n$; logical gates $L_1, \ldots, L_m$; and any number of dedicated, PP, or FCFS repair units, which satisfies the destruction by failure assumption. Each basic event B_i will have m_i failure modes $M_1, \ldots, M_{m_i}$.

Proposition 27. *Given the I/O-IMC semantics of an* ARCADE *model as above with basic events $B_1, \ldots, B_n$, we find that the spontaneous sets are a subset of*

$$\{\{f_{B_i}^{(M_j)}\} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\} \cup \{\{r_{B_i}\} \mid 1 \leq i \leq n\}.$$

Proof. This is a direct consequence of the fact that failure and repair actions are the only spontaneous actions in the I/O-IMCs describing the ARCADE elements and the spontaneous sets of a parallel composition of I/O-IMCs is a subset of the union of the sets of spontaneous sets of its components (see Theorem 53). □

Proposition 28. *Given the I/O-IMC semantics of an* ARCADE *model as above with basic events $B_1, \ldots, B_n$, we find that there is only one initial set which is*

$$\{\tau_{B_i} \mid 1 \leq i \leq n, B_i \text{ has an initial destructive operational state.}\}.$$

Proof. This is a direct consequence of the fact that the destruction actions are the only possible initial actions in the I/O-IMC semantics of the ARCADE elements and the initial sets of the parallel composition of I/O-IMCs equals the set of all possible combinations of the initial sets of its component I/O-IMCs (see Theorem 54). □

Recall that in Chapter 8 we have shown that any set of actions that is enabled simultaneously, must have a *common cause*, i.e., a set of actions that is either spontaneous or initial, and that indirectly triggers these actions (see Corollary 13). Our results concerning the spontaneous and initial sets now tell us that any case of non-determinism in an ARCADE model must be caused by either a single spontaneous failure, a single repair action, or a set of immediate destructions that were initially enabled. We will now study the triggering relation for the I/O-IMC semantics of ARCADE models to find out which actions can be indirectly triggered in this way.

9.5.3 Triggering relation

Taking into account the destruction by failure assumption (see Subsection 9.5.1), we find that the following actions trigger each other.

Proposition 29. *Given the I/O-IMC semantics of an* ARCADE *model as above with basic events $B_1, \ldots, B_n$, and logical gates $L_1, \ldots, L_m$, we find that the triggering relation is a subset of,*

τ_{B_i} *triggers* f_{B_i}, *if BC* B_i *has a destructive operational mode,* (9.18)

f_{B_i} *triggers* τ_{B_j}, *if BC* B_j *has a destructive dependency on Failure*(B_i), (9.19)

$f_{B_i}^{(M_x)}$ *triggers* τ_{B_j}, *if BC* B_j *has a destructive dependency on Failure*(B_i, M_x), (9.20)

f_{L_k} *triggers* τ_{B_j}, *if BC* B_j *has a destructive dependency on LG* L_k, (9.21)

f_{B_i} *triggers* f_{L_k}, *if Failure*(B_i) *is an input to LG* L_k, (9.22)

$f_{B_i}^{(M_x)}$ *triggers* f_{L_k}, *if Failure*(B_i, M_x) *is an input to LG* L_k, (9.23)

f_{L_l} *triggers* f_{L_k}, *if Failure*(L_l) *is an input to LG* L_k, (9.24)

r_{B_i} *triggers* u_{B_i}, *for every BC* B_i, (9.25)

u_{B_i} *triggers* u_{L_k}, *if Failure*(B_i) *or Failure*(B_i,M_x) *is an input to LG* L_k, *and* (9.26)

u_{L_l} *triggers* u_{L_k}, *if Failure*(L_l) *is an input to LG* L_k, (9.27)

where $1 \leq i, j \leq n$ and $1 \leq k, l \leq m$ and x is such that M_x is a failure mode of B_i.

Proof. This follows directly from the triggering relations of the I/O-IMC semantics of the individual ARCADE elements, and the fact that the triggering relation of a parallel composition of I/O-IMCs is a subset of the union of the triggering relations of its constituent I/O-IMCs (see Equation (8.5)). □

First of all, the destructive action τ_{B_i} of any BC with a destructive operational state triggers its failure (see (9.18)). Then, failure of a BC or a LG may trigger the destructive action of a BC, but only if there exists an appropriate destructive dependency (see (9.19) and (9.21)). From our discussion of BCs we know that mode-switching actions may trigger the destructive action τ_{B_i} and the destruction-by-failure assumption ensures that such mode-switching actions are always failure actions. Finally, we have that the failures of the inputs of an LG trigger the failure of the LG itself (see (9.22), (9.23), and (9.24)). In other words, the failure of a subsystem may be caused by the failure of one of its components. The repair of any BC triggers its recovery (see (9.25)). Finally, recovery actions of inputs will trigger the recovery actions of the logical gates they belong to (see (9.26) and (9.27)).

We can now discuss the *indirect triggering relation*. To do this, we will first introduce the *indirect dependency* relation on BCs and LGs of an ARCADE model.

Definition 116. *Given an* ARCADE *model with BCs $B_1, \ldots, B_n$ and LGs $L_1, \ldots, L_m$, we say that a BC or LG x* indirectly depends *on a BC or LG y if we can find a sequence of $z_1, \ldots, z_k$ of BCs and LGs such that $x = z_1$, $y = z_k$, and for each $1 < i \leq k$ we have that*

1. *z_i is a BC that destructively depends on BC or LG z_{i-1}, or*
2. *z_i is a LG that has BC or LG z_{i-1} as an input.*

Note that any BC or LG indirectly depends on itself (we find a sequence of length one).

We can see that the two conditions of Definition 116 are closely related to the triggering relation for ARCADE models. We will now use Definition 116 to describe the indirect triggering relation of an ARCADE model.

Theorem 64. *Given the I/O-IMC semantics of an* ARCADE *model as above (i.e., it satisfies the destruction by failure assumption), we find the that the indirect triggering relation is a subset of the union of the identity relation on all actions and*

$$
\begin{aligned}
&\{(\tau_{B_i}, \tau_{B_j}), (\tau_{B_i}, f_{B_j}), (f_{B_i}, \tau_{B_j}), (f_{B_i}, f_{B_j}),\\
&\qquad (f_{B_i}^{(M_x)}, \tau_{B_j}), (f_{B_i}^{(M_x)}, f_{B_j}) \mid B_j \textit{ indirectly depends on } B_i\}\\
&\cup \{(\tau_{B_i}, f_{L_k}), (f_{B_i}, f_{L_k}), (f_{B_i}^{(M_x)}, f_{L_k}) \mid L_k \textit{ indirectly depends on } B_i\}\\
&\cup \{(f_{L_l}, f_{L_k}) \mid L_k \textit{ indirectly depends on } L_l\}\\
&\cup \{(r_{B_i}, r_{B_j}), (r_{B_i}, u_{B_j}), (u_{B_i}, r_{B_j}), (u_{B_i}, u_{B_j}) \mid B_j \textit{ indirectly depends on } B_i\}\\
&\cup \{(r_{B_i}, u_{L_k}), (u_{B_i}, u_{L_k}) \mid L_k \textit{ indirectly depends on } B_i\}\\
&\cup \{(u_{L_l}, u_{L_k}) \mid L_k \textit{ indirectly depends on } L_l\},
\end{aligned}
$$

where $1 \leq i, j \leq n$ and $1 \leq k, l \leq m$, and M_x is a failure mode of B_i.

Proof. Theorem 64 follows from Proposition 29 by building the transitive, reflexive of the triggering relation for ARCADE models. □

To summarise, we see that failure and destruction actions of one BC or LG trigger failure and destruction actions of another BC or LG as long as the latter indirectly depends on the former. The same holds for recovery and repair actions. Note that Theorem 64 could be made preciser by making use of the fact that destructive dependencies do not influence repair and recovery actions, but it turns out this is unnecessary for our purposes. One important consequence of the above theorem is that failure and recovery are, in a sense, independent.

Lemma 22. *No recovery or repair action indirectly triggers a failure or destruction action. Neither does any failure or destruction action indirectly trigger a recovery or repair action.*

Proof. Lemma 22 follows directly from Theorem 64. □

9.5.4 Non-confluent pairs of actions

We now list the pairs of actions that can be non-confluent in the semantics of an ARCADE syntactical element.

Proposition 30. *The following pairs of actions can be non-confluent for the I/O-IMC semantics of a*

1. *basic component:*
   (a) $u_{B_i} \& \tau_{B_j}$, *where* B_j *destructively depends on* B_i,
   (b) $u_{L_i} \& \tau_{B_j}$, *where* B_j *destructively depends on* L_i,
2. *logical gate:*
   (a) $f_{L_i} \& u_{B_j}$ *and* $u_{L_i} \& f_{B_j}$, *when* $Failure(B_j)$ *is an input of* L_i,
   (b) $f_{L_i} \& u_{B_j}$ *and* $u_{L_i} \& f_{B_j}^{(F_x)}$, *when* $Failure(B_j, F_x)$ *is an input of* L_i,
   (c) $f_{L_i} \& u_{L_j}$ *and* $u_{L_i} \& f_{L_j}$, *when* L_j *is an input of* L_i,
3. *repair unit:*
   (a) $f_{B_i} \& r_{B_i}$ *and* $f_{B_i}^{(F_x)} \& r_{B_i}$, *for all BCs that have a repair unit,*
   (b) $f_{B_i} \& f_{B_j}$, $f_{B_i}^{(F_x)} \& f_{B_j}$, $f_{B_i} \& f_{B_j}^{(F_y)}$, *and* $f_{B_i}^{(F_x)} \& f_{B_j}^{(F_y)}$, *if distinct BCs* B_i *and* B_j *have the same FCFS repair unit.*

Proof. Proposition 30 is simply a restatement of the non-confluent pairs of actions we have found in Section 9.2. □

9.5.5 Sufficient conditions for determinism

We are now ready to find conditions that will ensure that the semantics of an ARCADE model are deterministic. These conditions will be based on the sufficient conditions for determinism of an I/O-IMC as described in Theorem 60. As for I/O-IMCs we will look for necessary conditions for non-determinism first, under the destruction by failure assumption.

Recall from Theorem 60 that we need three ingredients to have non-determinism in a distributed I/O-IMC. A pair of actions a and b that are either both initial or both in the same spontaneous set. These actions must then indirectly trigger two actions c and d respectively. Finally, actions c and d must be non-confluent for the I/O-IMC semantics of one of the syntactic elements of the ARCADE model. We have already seen with Lemma 22 that failure and recovery actions cannot (indirectly) trigger each other. We now extend this lemma to actions that satisfy Theorem 60.

Lemma 23. *Given the I/O-IMC semantics of an* ARCADE *model as above and four actions a, b, c, and d that satisfy Theorem 60 as above, we have that either*

1. *actions c and d are both destruction actions or failure actions (i.e., of the form τ_x, f_x, of $f_x^{(F)}$), or*
2. *actions c and d are both repair actions or recovery actions (i.e., of the form r_x or u_x).*

Proof. We prove Lemma 23 by contradiction. Assume then, without loss of generality, that action c is either a destruction or failure action, and action d is a repair or recovery action. Since the four actions a, b, c, and d satisfy Theorem 60 we have that a indirectly triggers c and b indirectly triggers d. Lemma 22 then gives us that a must also be a destructive or failure action and b in turn must be a repair or recovery action. To satisfy the first condition of Theorem 60 actions a and b must either both be initial or both part of the same spontaneous set. However, Proposition 27 tells us that there are no spontaneous sets which contain both a destruction/failure action and a repair/recovery action. Similarly, Proposition 28 gives us that there are no initial recovery or repair actions. This constitutes a contradiction and it follows that Lemma 23 holds. □

Lemma 23 greatly helps us in excluding many cases of non-confluence in ARCADE models as possible causes for non-determinism. Of all the pairs of non-confluent actions listed in Proposition 30 only one case satisfies Lemma 23. This is the case of the failure actions of two distinct BCs B_i and B_j which are both repaired by the same FCFS repair unit.

The third condition of Theorem 60 must then be satisfied by a pair of failure actions of BCs that are repaired by the same FCFS repair unit. The first condition, on the other hand, must be satisfied by either a single (spontaneous) failure action (see Proposition 27) or by two initially enabled destruction actions (see Proposition 28). To also satisfy the second condition of Theorem 60 we must find that the latter action or actions indirectly trigger the former. From Theorem 64 we know that we must then find

that the BCs that are repaired by the FCFS repair unit indirectly depend on the BC(s) that correspond to the initial/spontaneous actions. This finally gives us the following necessary conditions for non-determinism in an ARCADE model.

Theorem 65. *Given a well-formed* ARCADE *model that satisfies the destruction by failure assumption and which consists only of basic components, logical gates, and dedicated, PP, or FCFS repair units, we have that the I/O-IMC semantics of this* ARCADE *model is non-deterministic,* only *if the following conditions hold.*

1. *Either*
   (a) *there is a BC B_i, or*
   (b) *there are two BCs B_j and B'_j that both have an initial destructive operational mode.*
2. *Furthermore, there are two BCs B_k and B'_k that either*
   (a) *both indirectly depend on B_i, or*
   (b) *indirectly depend on B_j and B'_j, respectively, and*
3. *the* ARCADE *model contains a FCFS repair unit which repairs both basic components B_k and B'_k.*

Proof. The three conditions of Theorem 65 follow the three conditions of Theorem 60. From Propositions 27 and 28 it follows that the only way the first condition of Theorem 60 can be satisfied is through the first condition of Theorem 65. From Theorem 64 it follows that the only way for the action(s) from condition 1 to indirectly trigger the failure actions of two BCs B_k and B'_k (i.e., to satisfy the second condition of Theorem 60 is by satisfying the satisfying the second condition of Theorem 65. Finally, we have from Proposition 23 that the only way that the I/O-IMC semantics of an ARCADE model can satisfy the third condition of Theorem 60 (i.e., non-confluent actions) is to satisfy the third condition of Theorem 65. □

With Theorem 65 we have partly accomplished our goal of finding sufficient conditions to prove that an ARCADE model is deterministic. The sufficient conditions are of course that the ARCADE model conforms to the assumptions we have made and that it *does not* satisfy the three conditions of Theorem 65. That is, it only consists of basic components, logical gates, and the three types of repair units we have covered, it conforms to the destruction by failure assumption, and the BCs repaired by an FCFS repair unit are not indirectly dependent on the same BC (or different BCs that both have an initial destructive operational state).

9.5.6 Sufficient conditions for non-divergence

Since the presence of time-divergence in an I/O-IMC complicates its analysis and may be an indication of a modelling error, it will be useful to be able to show that the I/O-IMC semantics of an ARCADE model is non-divergent. In Chapter 8 we have found

Theorem 61 which tells us that the absence of "cycles" in the triggering relation of an I/O-IMC shows that the I/O-IMC is non-divergent. Again, we first give a necessary condition for the I/O-IMC semantics of an ARCADE model to be divergent.

Theorem 66. *Given an* ARCADE *model that satisfies the destruction by failure assumption and which consists only of basic components, logical gates, and dedicated, PP, or FCFS repair units, we have that the I/O-IMC semantics of this* ARCADE *model is divergent,* only *if the following condition holds. There exists a sequence* $z_1, \ldots, z_n$ *of length at least two, such that* $z_1 = z_n$ *and for each* $1 < i \leq n$ *we have*

1. z_i *is a BC that destructively depends on BC or LG* z_{i-1}*, or*
2. z_i *is a LG that has BC or LG* z_{i-1} *as an input.*

Proof. From our discussion of the triggering relation of an ARCADE model, we can see that the only way to have a cyclic triggering relation is to have a sequence of basic components and logical gates that cyclically depend on each other. Our condition is then simply a restatement of the definition of indirect dependence with the added condition that the sequence of BCs and LGs has length at least 2 and is a cycle. □

We then find a sufficient condition for the I/O-IMC semantics of an ARCADE model to be non-divergent, namely that the condition in Theorem 66 does not hold, i.e., there exists no cycle of destructively dependent BCs and LGs.

9.5.7 Spare management units

For the sake of brevity and simplicity, we have not discussed the I/O-IMC semantics of spare management units [3] in this chapter and they are not considered in Theorem 65. It should be noted that spare management units are indeed a likely cause of non-determinism [6]. We conjecture that for ARCADE models with spare management units another condition for non-determinism occurs when two spare management units share a spare C and each of these spare management units has another spare (A respectively B), such that spares A and B have a common cause (i.e., the two spares A and B play the same role as the two BCs repaired by the same FCFS repair unit in Theorem 65). In this scenario, the spares A and B may fail at the same time since they have a common cause) and then either spare management unit may "claim" the spare C. The decision which spare management unit actually claims spare C is then non-deterministic.

However, to fully understand the ways in which the use of spare management units may lead to non-determinism it is necessary to study the causality and confluence properties of the I/O-IMC semantics of spare management units as we have done for other ARCADE elements.

9.5.8 Algorithm and Complexity

We will now discuss how we can algorithmically ascertain whether an ARCADE model satisfies the conditions of Theorem 65. Below is a sketch of such an algorithm. We

assume that a concise description of the ARCADE model is given from which we can determine the principal relationships between the ARCADE elements, i.e., which basic components are repaired by which repair unit, on what basic components or logical gates does a certain basic component have a destructive dependency, etc., in constant time (for each component). Algorithm 2 checks whether an ARCADE model satisfies the conditions of Theorem 65.

noend 2 Checks whether an ARCADE model satisfies the conditions of Theorem 65. If the algorithm returns "True" then the ARCADE model may be non-deterministic. If it returns "False" then the ARCADE model is guaranteed to be deterministic.

```
1: Compute the set X of all pairs of BCs that are repaired by the same FCFS repair
   unit.
2: Compute the set Y of all BCs that have an initial destructive operational state.
3: Compute the relationship R on the set of all BCs and LGs that contains all pairs
   (x, y) such that x is an input of y or y destructively depends on x.
4: Compute the transitive reflexive closure R' of R; this relation describes the indirect
   dependencies between BCs and LGs.
5: for all (x, y) in X do
6:    for all BC z do
7:       if (z, x) and (z, y) are in R' then
8:          return True
9: for all (x, y) in X do
10:   b = False
11:   for all BC z in Y do
12:      if (z, x) is in R' then
13:         b = True
14:   if not b then
15:      return False
16:   for all BC z' in Y do
17:      if (z', y) is in R' then
18:         return True
19: return False
```

If a pair (x, y) is found in the loop 5-8 or the loop 9-18, then the ARCADE model may be non-deterministic. If not, it is guaranteed that the conditions of Theorem 65 do not hold and the I/O-IMC semantics of the ARCADE model must then be deterministic.

Theorem 67. *Given a well-formed* ARCADE *model consisting of basic components, logical gates, and dedicated, FCFS, and PP repair units which satisfies the destruction-by-failure assumption, if Algorithm 2 returns "False" then the* ARCADE *model is deterministic*

Proof. We prove Theorem 67 by contradiction. Assume then that we have an ARCADE model as in Theorem 67 which is *non-deterministic* and for which Algorithm 2 returns "False".

Since our ARCADE model is non-deterministic, we have that it satisfies the conditions in Theorem 65. We first consider the case that the ARCADE model has a BC B_i and two BCs B_k and B'_k which are both indirectly triggered by B_i and are both repaired by the same FCFS repair unit. It is immediately clear that the pair (B_k, B'_k) is in the set X computed in step 1 of the algorithm. Furthermore we have that, since B_i indirectly triggers both B_k and B'_k that the pairs (B_i, B_k) and (B_i, B'_k) are in the relationship R' computed in step 4 of the algorithm. We then have that in the loop 5-8 of the algorithm we will find the case $(x, y) = (B_k, B'_k)$ and $z = B_i$ and in this case the condition of step 7 will be satisfied, which will mean the algorithm returns "True" and this is a contradiction.

We now consider the remaining possibility, that there are two BCs B_j, B'_j which both have a destructive initial operational mode and two BCs B_k and B'_k as above which are indirectly triggered by B_j respectively B'_j. Again we have that the pair (B_k, B'_k) is in the set X and both B_j and B'_j are in the set Y computed in step 2 of the algorithm. Finally we have that the pairs (B_j, B_k) and (B'_j, B'_k) are in the relationship R'. Now, for the loop 9-18, consider the case that $(x, y) = (B_k, B'_k)$ and for the loop 11-13 consider the case that $B_j = z$. We then have that the condition of step 12 holds and b will be "True" when we reach step 14 in this particular iteration of the loop. For the loop 16-18 then consider the case where $B'_j = z'$. We then have that the condition in step 17 holds and the algorithm will again return "True" leading to a contradiction. □

In both time and space complexity step 4 of this algorithm dominates the other steps. This step has cubic complexity (finding the transitive closure of a relation) in the number of BCs and LGs of the ARCADE model being studied. This is significantly less than the space and time complexity of constructing the minimised I/O-IMC semantics of the ARCADE model which is exponential in the number of syntactical elements in the ARCADE model [13].

We can check for the possibility of time-divergence in a similar way. However, in step four of the algorithm we should compute the transitive closure instead of the transitive reflexive closure. It then suffices to check if this transitive closure contains (X, X) for any BC or LG X. If so, the I/O-IMC semantics may be divergent. If not, we can be sure that the I/O-IMC semantics of the ARCADE model is non-divergent. Again we find cubic time and space complexity in the number of basic components and logic gates of the ARCADE model.

The algorithms presented in this section greatly improve the complexity of verifying whether an ARCADE model is deterministic and non-divergent. Instead of having to construct and inspect the exponentially large state space of the I/O-IMC semantics of the ARCADE model, we can now verify for many ARCADE models that they are deterministic and non-divergent by using our algorithms which are cubic in the number of syntactical elements of the ARCADE model and do not require constructing its I/O-IMC semantics. It is important to note that the algorithms may give rise to false negatives. That is, there may be ARCADE models which are deterministic, but which still satisfy the conditions in Theorem 65 and thus are not recognized by Algorithm 2 as being non-deterministic. However, to counteract these false negatives it may be possible

to partially construct the I/O-IMC semantics of the ARCADE model to show that the combination of basic components that satisfy the conditions of Theorem 65 is spurious.

9.6 Discussion

In this chapter, we have studied the I/O-IMC semantics of dependable systems modelled using ARCADE. We have first presented a translation of ARCADE syntactical elements to I/O-IMCs based on the work of Maaß [34]. The semantics of ARCADE is compositional in a very deep sense: It directly arises from the parallel composition of the semantics of its syntactical components, which by virtue of Theorems 38 and 39 is modular.

We have further given sufficient conditions for the I/O-IMC semantics of an ARCADE model to be deterministic. This has only been possible because of the results of Chapter 8. In order for an ARCADE model to be non-deterministic a very specific condition must hold, namely that there is a FCFS repair unit with two BCs that, directly or indirectly, destructively depend on the same BC or LG. This condition can be checked very efficiently using Algorithm 2 and in Subsection 9.6.1 we will discuss how this algorithm can be used to make the analysis of ARCADE models more effective. In this chapter we did not consider ARCADE models with spare management units. For the full ARCADE formalism, including spare management units, sufficient conditions for determinism can be established in a similar way, but it will be more challenging as SMUs have a complex semantics and may be a cause of non-determinism in ARCADE models.

9.6.1 Analysis of Arcade models

We now consider how we might *analyse* important properties of an ARCADE model. Recall that system properties are modelled by using logical gates. This means that a property correspond to a certain subset of ARCADE components being operational or in a particular failure mode. For instance, in our "pump system" example from Section 9.1.8 we modelled that the "no cooling" property holds when both pump lines are blocked. In our "replicated web server" example, the "service unavailable" property holds when all three servers are inoperational.

In this subsection we will discuss how we can compute interesting measures for these properties, such as:

- What is the probability that the pump system is not cooling after 2 months?
- What fraction of time can the replicated web server be expected to be available in the long run?
- What is the probability that the pump system stops cooling at least once in the first year of operation?

We will first look at different classes of ARCADE models before discussing more advanced solution algorithms to improve the efficiency of analysis.

Analysis of closed Arcade models

Theorem 63 tells us that the I/O-IMC semantics of a closed ARCADE model is a complete I/O-IMC, i.e., an I/O-IMC with no output or input actions. In Chapter 7 we have shown that any closed I/O-IMC can be interpreted as a CTMDP. We can thus use standard CTMDP analysis techniques to compute interesting measures for the ARCADE model, such as the algorithm by Neuhäusser and Zhang [38] as discussed in Subsection 7.7.2.

Analysis of deterministic closed Arcade models

The operational behaviour of a *deterministic* closed ARCADE model is a deterministic complete I/O-IMC. Theorem 52 establishes that such I/O-IMCs can be interpreted as CTMCs. We can use standard CTMC solution techniques, such as simulation [19], uniformisation [46], and fast adaptive uniformisation [15] to compute interesting measures such as the probability of being in a "no cooling" state at a particular point in time or the long-run average probability of being in a "system unavailability" state.

Compositional minimisation

Before we can analyse an ARCADE model we must in general first construct the I/O-IMC that represents its operational behaviour. We have seen in Section 9.2.6 that this I/O-IMC is the parallel composition of the I/O-IMCs that represent the syntactic elements of the ARCADE model. However, the size of this parallel composition grows exponentially in the size of the ARCADE model. We will now see that we can mitigate this problem by applying the technique of *compositional minimisation*.

The I/O-IMC that represents the entire ARCADE model is simply the parallel composition of the I/O-IMCs that represent the ARCADE elements that make up the model. However, the size of this I/O-IMC grows exponentially in the number of syntactic elements, so even for moderately large ARCADE models it is infeasible to construct this I/O-IMC directly. Instead, *compositional minimisation* (see, e.g., [25]) is used to construct a smaller I/O-IMC that is weakly bisimilar to the parallel composition of the component I/O-IMCs. Given a set M of component I/O-IMCs, compositional minimisation proceeds as follows

1. select a subset of I/O-IMCs $M' \subset M$,
2. construct the parallel composition P of the I/O-IMCs in M' and hide all actions that are internal to this subset of I/O-IMCs,
3. compute the weak bisimulation minimisation P' of P, and
4. continue from step 1 with the set $(M \setminus M') \cup \{P'\}$ of I/O-IMCs, replacing the set M' by its minimised parallel composition P', until only one I/O-IMC is left.

Compositional minimisation avoids the construction of the parallel composition of all I/O-IMCs by applying weak bisimulation minimisation early and often. The efficacy of

this approach depends on the order in which the I/O-IMCs are composed and minimised. In other words, it depends on the selection of I/O-IMCs in step 1 above.

In [13] and [14] we have developed a heuristical algorithm that selects a good composition ordering, i.e., a composition ordering that allows us to construct the semantics of the ARCADE model both quickly and, more importantly, using little space to store the intermediate I/O-IMC models. The algorithm is based on trying to select, each time we reach step 1 of compositional minimisation, a set of I/O-IMCs in such a way that their parallel composition can be reduced a lot by weak bisimulation minimisation. The algorithm is further based on two assumptions

1. the number of transitions that can be eliminated due to weak bisimulation minimisation is proportional to the relative number of reachable internal transitions in an I/O-IMC, and

2. the number of reachable internal transitions in an I/O-IMC is proportional to the total number of internal transitions including those that are unreachable.

The total number of, reachable and unreachable, internal transitions can be computed efficiently from the transition relations of the components I/O-IMCs [14].

The heuristic used by the algorithm to select the subset of I/O-IMCs to compose is based on this estimated proportion of internal transitions, on an estimation of the amount of *interleaving* in the parallel composition (see [14]), and on the number of I/O-IMCs in the parallel composition. The algorithm then aims to select that subset of I/O-IMCs that has the highest value for this heuristic.

The result of the compositional minimisation algorithm is a complete I/O-IMC that is weakly bisimilar to the parallel composition of the set of I/O-IMCs we started with. To be exact, this I/O-IMC is the smallest such I/O-IMC(with respect to the number of states).

Given an explicit representation of the minimised I/O-IMC semantics of the ARCADE model it is easy to determine whether the I/O-IMC is deterministic or not. If the I/O-IMC has any interactive transition (which must necessarily be internal), then the I/O-IMC is non-deterministic, otherwise it is deterministic. Depending on whether the I/O-IMC is non-deterministic or not, we can then transform it into an equivalent CTMDP or CTMC, respectively and analyse it using standard solution techniques.

In general, the size of the minimised I/O-IMC computed using compositional minimisation still grows exponentially in the number of component I/O-IMCs. In the next subsection we will try to avoid generating this possibly prohibitively large I/O-IMC altogether.

On-the-fly techniques

We can use the results of this chapter to efficiently analyse ARCADE models by avoiding the construction of the entire state space of the I/O-IMC representing an ARCADE model. We can achieve this by applying *on-the-fly* solution techniques.

On-the-fly solution techniques compute interesting measures for a model without constructing its entire state space. Instead, transitions of the model are generated on-the-fly, as required by the solution technique. Examples of on-the-fly solution techniques for CTMCs are simulation [19] and fast adaptive uniformisation [15]. The advantage of on-the-fly techniques is that we do not need to generate or store the entire state space of the I/O-IMC that represents the ARCADE model. Instead, we work with the I/O-IMC representations of the syntactical elements of the ARCADE model.

The down-side of such on-the-fly techniques is that they can only be applied to CTMCs, which means that they can only be applied to deterministic I/O-IMCs. Traditionally, the only way of knowing that an I/O-IMC is deterministic is to build its state space [3]. However, this defeats the purpose of on-the-fly solution techniques, which is to avoid building the entire state space explicitly. Fortunately, Algorithm 2 allows us to show that an ARCADE model is deterministic efficiently and without constructing the entire state space of the underlying I/O-IMC.

Given a deterministic ARCADE model, we know that its I/O-IMC semantics is deterministic, and Theorem 52 then gives us that any behaviour of this I/O-IMC results in the same CTMC. In order to use on-the-fly solution techniques we must be able to generate this CTMC on-the-fly. That is, given a particular state we must be able to compute the set of outgoing transitions for this state. We can easily compute outgoing transitions for the I/O-IMC by applying the definition of parallel composition for I/O-IMCs. To construct transitions of the underlying CTMC we can then simply pick an arbitrary scheduler to resolve the "non-deterministic" choices, since we know that these choices will always lead to weakly bisimilar states.

Maaß has shown that simulation can be used to study deterministic ARCADE models [34], although simulation is not very effective for *stiff* models, i.e., models with both very large and very small stochastic rates (i.e., repair and failure rates). Unfortunately, most ARCADE models are indeed stiff as repair rates are usually much larger than failure rates. An interesting alternative is fast adaptive uniformisation, which is more effective for stiff CTMCs [15]. However, fast adaptive uniformisation can, for the moment, only be used to compute *transient* measures for a CTMC, i.e., we can only use it to answer questions such as "what is the probability of being in a service unavailable state at time x?"

Combining compositional minimisation with on-the-fly techniques

Finally, we propose that a combined approach may be used to analyse deterministic ARCADE models that cannot be analysed using compositional minimisation and standard solution techniques. By first performing as many compositional minimisation steps as possible, we can exploit as many symmetries as we can to reduce the size of the underlying CTMC. When further compositions become infeasible due to time and space restrictions, we can attempt to apply fast adaptive uniformisation to the smaller set of I/O-IMCs generated by the compositional minimisation steps.

9.6.2 Other measures

So far we have focused on computing *instantaneous unavailability* properties of an ARCADE model, i.e., the probability that a certain system property holds at a certain time. Of course, there are many other interesting metrics that can be computed for a dependable system. Here, we review a number of them and discuss how these metrics fit into the ARCADE framework. Note that, the question of which measure to compute is in general orthogonal to the problem of how to compute or induce the state space of the I/O-IMC semantics of an ARCADE model as we discussed in the previous subsection. We will discuss several different kinds of measures that are particularly interesting in the context of ARCADE. For a general overview of verification algorithms for CTMDPs we refer to Buchholz et al as a starting point [9].

Reliability.

The *reliability* of a system is the probability that the system will not fail before a given time instance T. Different from availability, reliability is not concerned with the state of the system at time T but the number of failures *before* time T. In other words, reliability is the probability that zero failures occurred before time T. We can measure the reliability of an ARCADE model using state labels. To do this, we add an I/O-IMC to the semantics of the ARCADE model which "counts" the number of failures of the ARCADE model. This I/O-IMC has as input action the actions f_s which describes the failure of the complete system (this action usually corresponds to the failure action of a logical gate that describes the failure condition for the complete system). The I/O-IMC has no other actions, but one of its states is labelled "at least one failure". Figure 9.13 shows this I/O-IMC.

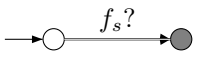


Figure 9.13: Reliability I/O-IMC (left) that counts the number of failures of a dependable system. The state that is labelled "at least one failure" is grey.

As in the case for availability we have that the state-labels of a composed state are found by the union of the state-labels of its component states. It is clear then that, every state of the complete I/O-IMC semantics of the ARCADE model that is labelled "at least one failure" corresponds to a state where the system has indeed failed at least once. We can compute the reliability of the system as the probability that the I/O-IMC semantics of the ARCADE model does not occupy a state labelled with "at least one failure" at time T. Obviously, this technique can also be applied to compute properties such as "at least two failures at time T, and so forth, by altering the counting I/O-IMC in Figure 9.13.

Cost-based measures

Both availability and reliability tell us something about the probability that certain events happen for the dependable system being studied. A different type of metrics is based on *costs* or *rewards*. For instance, we may want to compute how much money we can expect to need to repair our dependable system. In ARCADE we can model costs simply by associating costs to the states of the I/O-IMC semantics of the relevant ARCADE syntactical models.

For instance, to model the cost of repairs, we can associate an appropriate cost (per time unit) to those states of the repair units where the RU is actively repairing a basic component. Of course, we can use different costs for different RUs. We may also want to model the *revenue* of a system when it is operational. Assume we have a logical gate that models the failure condition of the system. We can then associate a reward with the states of this logical gate that are operational to model the revenue of the system. Of course, if we model both costs and rewards we must make sure that the costs are negative and rewards are positive.

As we have seen in Section 2.1 we can use costs/rewards to induce the state equivalence relation $=_s$. That way we ensure that any two states that are equivalent with respect to $=_s$ (and then also any two states that are equivalence with respect to weak bisimulation) have the same cost/reward. The cost of a composed state is the sum of the costs of its constituent states.

The semantics of the complete ARCADE model is then an I/O-IMC decorated with rewards. Similar to I/O-IMCs with state labels, such I/O-IMCs can be interpreted as CTMDPs with state-rewards or, when the I/O-IMC is weakly deterministic, CTMCs with state-rewards (also known as Markov reward models (MRMs)). There exist various solution techniques to compute interesting cost/reward metrics for such models. We list a number of these techniques here. Buchholz and Schulz have shown that we can compute both the maximal/minimal cumulative reward over a finite time horizon for CTMDPs [10]. Finally, we have that the Markov reward model checker (MRMC), CADP, and the PRISM tool support various cost and reward based metrics for CTMCs and CTMDPs [31, 32, 18].

Long-run properties of Arcade models

It is also interesting to consider long-run properties of ARCADE models, that tell us something about the entire life-time of the dependable system. For instance, the *long-run average unavailability* of a dependable system is the fraction of time the system is inoperable over an infinitely long time-span. In other words, it is the fraction of time the system is down over the time-interval $[0, T)$ where T goes to infinity.

For CTMCs, we can compute the long-run unavailability by computing the probability that the system is down at time T where T goes to infinity [46]. The probability distribution at time T, where T goes to infinity is called the *limiting distribution*. Most stochastic model-checking tools can be used to compute this distribution. For CTMDPs the infimum/supremum long-run average unavailability can be computed using the long-run average expected reward algorithm by Wimmer et al. [50]. This is accomplished by

assigning a reward of one to those states of the ARCADE semantics where the system is down and then computing the long-run average reward.

10 Conclusion

In this thesis, we have studied I/O-IMCs as a combination of the non-deterministic and compositional formalism of input/output automata (see Chapter 4) and the stochastic model of continuous-time Markov chains (see Chapter 3). To conclude this thesis we will highlight several key results and propose avenues for future research.

10.1 Modular semantics

We have seen in Chapter 3 that a simple matrix (the infinitesimal generator matrix) can be seen as a syntactical representation of a CTMC. If we assume different CTMCs are independent, then this semantics is modular: two generator matrices can be composed through interleaving and their CTMC semantics arises as the independent combination of the two CTMCs corresponding to the generators. In Chapter 4 we saw that our variant of IOA also has a modular semantics: the fair reach-traces of a composite IOA can be dissected to find the fair reach-traces of its component IOA.

We have then defined the semantics of I/O-IMC as an orthogonal combination of the semantics of CTMCs and IOA. We use interactive jump processes (introduced in Chapter 6) as the semantic underpinning of I/O-IMCs. Crucially, the jumps of interactive jump processes consist of two parts: a Markovian jump, which is governed by the rules of CTMCs and whose jump probabilities are determined by the Markovians transitions of the I/O-IMC, and an interactive jump, which is governed by the rules of IOA and whose interactions are determined by the interactive transitions of the I/O-IMC. The Markovian and interactive jumps do not interfere with each other because we assume the interactive jumps to be instantaneous (via the well-known maximal progress assumption [39]). As for CTMCs and IOA, we have that the semantics of I/O-IMCs is modular: an interactive jump process describing the stochastic behaviour of a composite I/O-IMC can be decomposed to find the stochastic behaviour of its components.

In Chapter 9 we used I/O-IMCs to provide a compositional operational semantics for ARCADE models which can be used to describe dependable systems. This is an example of *triple compositionality*. Figure 10.1 illustrates the point. ARCADE itself is compositional as we can combine different syntactical elements to create an ARCADE model. In Chapter 9 we have then given an operational I/O-IMC semantics to each of the syntactical elements of the ARCADE model. The I/O-IMC semantics of the complete ARCADE model then arises naturally from the composition of the I/O-IMCs representing its constituent parts. However, we can go even deeper than that. In Chapter 6 we have given a stochastic semantics to I/O-IMCs themselves in terms of sets of interactive jump processes. This gives us a modular stochastic semantics for ARCADE models in terms of interactive jump processes. In Section 9.3 we have seen that this allows us to reason about the stochastic properties of partial ARCADE models.

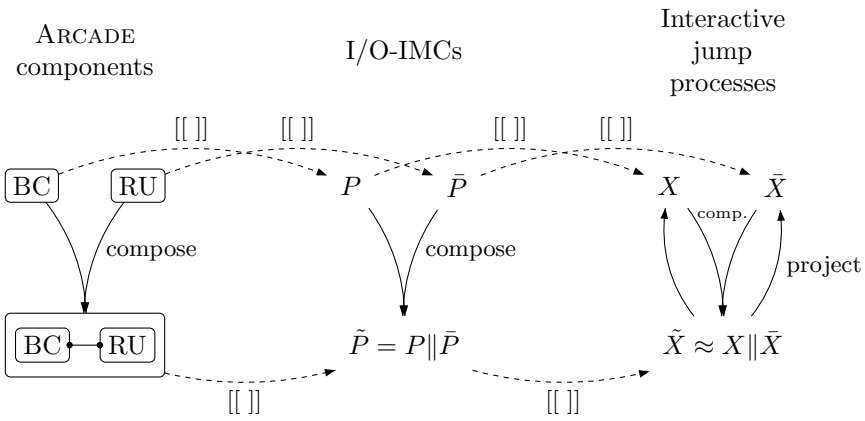


Figure 10.1: Illustration of triple compositionality. Arrows labelled [[ ]] point from a model to its semantics.

10.2 Dealing with non-determinism and divergence

IOA are inherently non-deterministic, since a single IOA may has as its semantics a set of traces and the choice between these traces is non-deterministic. This non-determinism is inherited by I/O-IMCs in the sense that the outcome of an interactive jump made by an I/O-IMC may be non-deterministic (since interactive jumps behave like IOA). In Chapter 6 we have used schedulers to characterize the resolution of these non-deterministic choices.

There are different ways of dealing with this non-determinism. We may put restrictions on our models to ensure non-determinism does not arise. This is for instance achieved for Wu-PIOA [53] by strictly interleaving Markovian and interactive transitions. Note that if we consider the subset of I/O-IMCs with strictly interleaved Markovian and interactive transitions it is clear to see that this indeed also avoids non-determinism. Another way of avoiding non-determinism is to add additional information to the model

which can be used to resolve non-deterministic choices. This approach is used by Cheung and others to avoid some forms of non-determinism in their work on switched probabilistic IOA [11].

In this thesis we have decided not to avoid non-determinism as it can represent interesting aspects of a system, such as the fact that there is missing information or the presence of unpredictable external influences which cannot be accurately modelled [44]. Instead of avoiding non-determinism, we try to understand why non-determinism arises. We also mitigate the effects of non-determinism by on the one hand offering easy to check sufficient conditions for a composite I/O-IMC to be deterministic in Chapter 8 and on the other hand showing that non-deterministic I/O-IMCs can be analysed using standard CTMDP solution techniques. When applied to the ARCADE formalism introduced in Chapter 9 we see that these sufficient conditions translate to an algorithm with cubic complexity in the size of the *syntax* of an ARCADE model. This algorithm can be used to efficiently verify that an ARCADE model is deterministic. We have discussed in Section 9.6 that the ability to efficiently ascertain that certain composite I/O-IMCs are deterministic could make it possible to analyse much larger composite I/O-IMCs than can be analysed with traditional analysis techniques that rely on constructing the entire state space of the composite I/O-IMC.

10.3 Avenues for future research

We now discuss several possible avenues of future research that build on the results presented in this thesis.

10.3.1 Modular schedulers

We have seen that there is an important caveat to be made with respect to the modularity results in Chapter 6. In Section 6.4 we have used *schedulers* to characterize the different behaviours of an I/O-IMC. We saw that these schedulers resolve non-determinism based on the history of the process in question. In Section 6.5.2 we were able to combine the schedulers of two component I/O-IMCs to find schedulers for their composition, but only when the component schedulers were able to reason about the history of the *composite* I/O-IMC and not just the component I/O-IMCs. In essence, the non-deterministic choices in a component I/O-IMC would be resolved by considering the history of all components in the composition.

As Giro and D'Argenio have pointed out, such schedulers are in many cases unrealistic [20]. A component of a system may be expected to resolve its non-deterministic choices based only on its own observations (i.e., its own history) not on the observations of the entire system. It will be interesting to see if we can find, in the context of I/O-IMCs, a class of schedulers similar to the distributed schedulers of Giro and D'Argenio, which overcomes this issue (albeit yielding undecidability results). Unfortunately, there are two major concerns when it comes to distributed schedulers. First, they negate the modularity of our semantics in a new way, because an I/O-IMC that arises through

a parallel composition will have a different set of schedulers (and thus a different semantics) than the same I/O-IMC regarded as an "atomic" I/O-IMC. This is caused by the fact that the schedulers of composed I/O-IMCs are restricted, but the schedulers of atomic (i.e., not composed) I/O-IMCs are not. The second issue with distributed schedulers is that we still need an appropriate way to resolve the non-determinism between different components (as opposed to non-deterministic choices within a component).

To deal with the first problem (a composite I/O-IMC $\tilde{P} = P \| \bar{P}$ allowing fewer schedulers than a monolithic I/O-IMC P' with the same states, actions, and transitions), we will have to add information to the composite I/O-IMC that conveys how non-deterministic choices may be resolved. For instance, when composing two I/O-IMCs P and $\bar{P}$ where P has output actions a and b and input action c, the composition of these I/O-IMCs must record the fact that non-deterministic choices between a and b may only be resolved by looking at when a-, b-, and c-events occurred, not by looking at the occurrence of any other actions. This information can be recorded by maintaining a set of sets of actions. For each set of actions in the set we have that a non-deterministic choice between two actions in the set may only be resolved by considering the time the actions in that set occurred. Each set of actions then corresponds to the actions that can be observed by one of the components of the composite I/O-IMC.

The second problem, how to resolve non-determinism between different I/O-IMCs (i.e., a composite I/O-IMC reaches a state where two actions controlled by different component I/O-IMCs are enabled), could be solved (in a somewhat unsatisfactory way) by adding a scheduler to each I/O-IMC which assigns a positive real number to an enabled action given that the I/O-IMC has followed a particular path. When composing such I/O-IMCs together the non-determinism between their actions can be resolved by assigning probabilities proportional to the value selected by our new scheduler. This approach has some down-sides, however. In particular, there is no natural scale for the values assigned by the new scheduler, since determining whether such a value is large or small only makes sense when considering other I/O-IMCs. Additionally, there seems to be no good intuitive meaning for these values.

10.3.2 Analysis of infinite-state I/O-IMCs

We have seen in Chapter 7 that I/O-IMCs can be analysed using standard CTMDP analysis techniques and, in the case of deterministic I/O-IMCs, CTMC analysis techniques. We have so far restricted our attention to the analysis of finite I/O-IMCs. However, there exist several techniques to analyse the transient properties of infinite-state CTMCs, which means that we can consider analysing infinite-state deterministic I/O-IMCs. In the literature we find several different approaches to the transient analysis of infinite-state CTMCs.

- We may simulate the stochastic behaviour of the CTMC up to a certain time-point (see e.g., Gillespie [19]), by generating trajectories and applying statistical techniques to approximate its stochastic properties,
- We may use the fact that, for any time-point t and threshold $\epsilon > 0$ we may

find a jump-index k such that the number of jumps exhibited by the CTMC in t time-units is less than k with probability greater than $1 - \epsilon$. In other words,

$$\Pr(J_k > t) > 1 - \epsilon.$$

This allows us to restrict our attention to the set of states that can be reached from an initial state with at most k jumps. Assuming the set of initial states is finite and the CTMC is finitely branching, this set of states will also be finite. The jump times of the CTMC are usually over-approximated using a Poisson process [36, 45, 21] or a birth process [15].

- Finally, we may restrict our attention to infinite-state CTMCs with a particular structure which allows us to efficiently analyse it (see e.g., Remke [43]).

Let us consider the analysis of a composite I/O-IMC $C = (P_1 \| \dots \| P_n) \backslash B$, where n is finite, but one or more of the component I/O-IMCs has countably infinite states. We will briefly discuss the following problems when trying to analyse infinite-state I/O-IMCs: how to find a finite representation for infinite-state I/O-IMCs, how to ascertain that a composite infinite-state I/O-IMC is deterministic and non-divergent, how to derive a CTMC from a closed infinite-state I/O-IMC, and how to ascertain that a composite infinite-state I/O-IMC is regular (or more accurately, corresponds to a regular CTMC).

Finite representation. First of all, we will need a finite way of representing our infinite-state I/O-IMCs. The most common way to do this is to use a symbolic representation of each state. For instance, a state may describe the number of people in an unbounded queue or the number of molecules of a certain chemical in a cell. Each state then has a finite number of transitions which may depend on the state itself (e.g., a certain transition may only be present when more than x people are in the queue or the value of a transition rate may depend on the number of molecules). This allows us to compute the outgoing transitions for any concrete state. In a similar way we can compute the outgoing transitions of the parallel composition of I/O-IMCs with a symbolic state space. Now we can explore the reachable states and transitions of the composite I/O-IMC by "unrolling" transitions as necessary.

Determinism. All the analysis techniques we described above operate on CTMCs, so they can only be applied to *deterministic* I/O-IMCs. This means we will need to determine whether our composite infinite-state I/O-IMC is deterministic. If the set of actions in our composite I/O-IMC is finite, then Algorithm 1 can be applied without any problem. If the number of actions is infinite (e.g., the actions have a symbolic representation as well) then we might still be able to show that the sufficient conditions for determinism (see Theorem 60) hold by using theorem proving (see e.g., Paulson [41]). The same considerations apply to the problem of figuring out if the infinite-state I/O-IMC is non-divergent.

From I/O-IMC to CTMC. We have seen in Chapter 7 that we can derive a CTMC from a deterministic I/O-IMC by choosing an arbitrary scheduler. We can find the outgoing CTMC-transitions of a stable state x of our I/O-IMC by considering all outgoing Markovian transitions of x and then unrolling interactive transitions until we reach another stable state. However, we must be careful to consider only fair traces (i.e., we may not "ignore" any set of transitions indefinitely). This can be achieved by deciding probabilistically which enabled interactive transition to take. It can be shown that the probability of selecting an unfair trace in this way is zero.

Regularity. We now have a symbolic representation of a CTMC (i.e., we can compute the outgoing transitions for any concrete state of the CTMC). However, all the approaches we consider can be used only for *regular* CTMCs (since non-regular CTMCs may make infinitely many jumps in a finite amount of time). It is then important to be able to show that our composite deterministic I/O-IMC C, corresponds to a regular CTMC. Recall from Subsection 3.1.5 that we can show regularity for an infinite-state CTMC in several ways. First of all, if the exit-rates of the CTMC are bounded from above then the CTMC is regular. It is clear that if the transition rates in the component I/O-IMCs are bounded then the transition rates of C and its underlying CTMC are also bounded. let If we cannot find a bound on the exit rates, we can try to find a Lyapunov function whose expected rate of change is linear (see Lemma 2 for details). We then have to find an appropriate Lyapunov function over the states of C which shows the regularity of the corresponding CTMC. Ideally, we would define this Lyapunov function compositionally. We would specify Lyapunov functions for all the components of C and provide a way to derive the Lyapunov function of a parallel composition from the Lyapunov function of its components. We must then make sure that the component Lyapunov functions satisfy certain properties which are preserved by parallel composition and hiding and which ensure regularity for the CTMC corresponding to C.

In the end, if we can show that our composite infinite-state I/O-IMC C is:

- deterministic,
- non-divergent, and
- regular,

then we can apply any of the infinite-state analysis techniques (with the exception of the structure-based approaches) to its underlying CTMC to compute transient properties for this I/O-IMC.

10.3.3 Analysis of open I/O-IMCs

Up until now, our analysis of I/O-IMCs has been restricted to *closed* I/O-IMCs, which do not interact with their environment. By giving a modular semantics to I/O-IMCs in Chapter 6 we have the possibility of analysing open I/O-IMCs. Such an open I/O-IMC represents a system that may still interact with its environment. We have shown in Section 9.3 that we could study a subset of the components of a composite I/O-IMC

which happens to be closed. But can we also study I/O-IMCs that are *open* (i.e., that may still be influenced by their environment)?

As an example, let us look at our pump system from subsection 9.1.8, but now consider the case that we simply do not know the dynamics of one of its valves (which we will call valve A). We could then attempt to analyse the I/O-IMC semantics of the pump system without this valve. Obviously, this I/O-IMC would be an open I/O-IMC with input actions $f_A^{(M)}$ and u_A, where M is the unique failure mode of the valve.

Theorem 35 gives us a formula to recursively compute the probability of reaching a particular state in an I/O-IMC after a certain number of jumps, where the non-determinism in the I/O-IMC is resolved through a pair of schedulers (the interactive jump scheduler γ which controls the interactive jumps of the I/O-IMC and the external jump scheduler η which controls the likelihood of an external jump). It will be very interesting to see if we can use this theorem to develop a way to compute bounds for the transient probabilities of an open I/O-IMC, in the same way as we compute bounds for the transient probabilities of a CTMDP (or closed I/O-IMC), where we only have to deal with the interactive jump scheduler.

One of the main challenges will lie in the fact that the external jump scheduler η is used to select a jump-rate, rather than a jump-probability (as for the interactive jump scheduler γ), which means that the values that η takes on are in principal unbounded. For our example, $\eta_\sigma^{(t)}$ represents the rate at which either of the input actions $f_A^{(M)}$ or u_A occurs at time t and after observing the path σ. In many cases, it may be optimal (to reach some state x) for these actions to occur as fast as possible, which means the valve either fails immediately or recovers from failure immediately. This means that we must choose $\eta_\sigma^{(t)}$ to be as large as possible, but this is problematic since η is unbounded and jump-rates in a CTMC are assumed to be finite.

A Proofs

A.1 Proofs of Chapter 6

A.1.1 Proof of Proposition 18

Given a stable interactive jump process X with state space S, actions A, and a probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ on the timed-paths of X, where $\mathcal{P}$ is an arbitrary probability function on $\mathcal{F}_{S,A}$, the following events are measurable.

1. For any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $w_i \in \mathcal{L}^V$, the set of trajectories where the i-th interactive jump starts in x_i, ends in y_i and has sequence w_i,
$$\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\},$$
is measurable.

2. For any time-points $t, s \in \mathbb{R}_{\geq 0}$ we have, that the set of trajectories where the first jump after time t occurs before time $t + s$,
$$\{\omega \mid J_1^{(t)}(\omega) \leq t + s\},$$
is measurable.

3. For any time-point $t \in \mathbb{R}_{\geq 0}$ and any state $x \in S_\perp$ we have, that the set of trajectories where the stochastic process X_{post} occupies state x at time t,
$$\{X_{\mathsf{post}}^{(t)} = x\},$$
is measurable.

Proof. First, we consider the event that the i-th interactive jump of the interactive jump process was a jump from a state x_i to a state y_i with action-sequence w_i. We find

$$\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\} = C'_H,$$

where

$$H = \{(x_0, w_0, y_0, t_1, \ldots, t_i, x_i, w_i, y_i) \mid \\ x_0, y_0, \ldots, x_{i-1}, y_{i-1} \in S_\perp, w_0, \ldots, w_{i-1} \in \mathcal{L}^V, t_1, \ldots, t_i \in \mathbb{R}_{\geq 0}\}.$$

For the second event we find

$$\{\omega \mid J_1^{(t)}(\omega) \leq t + s\} = \bigcup_{i=1}^{\infty} C'_{H_i},$$

where

$$H_i = \{(x_0, \ldots, t_{i-1}, x_{i-1}, w_{i-1}, y_{i-1}, t_i, x_i, w_i, y_i) \mid \\ x_0, \ldots \in S_\perp, w_0, \ldots \in \mathcal{L}^V, t_{i-1} \leq t < t_i \leq t + s\}.$$

We can combine the above two events to find the set of trajectories where X_{post} occupies a state $x \in S_\perp$ after the n-th jump, which is the last jump before time $t \in \mathbb{R}_{\geq 0}$. I.e., we have that the event

$$\{\omega \mid X_{\mathsf{post}}^{(J_n)}(\omega) = x, J_n(\omega) \leq t < J_{n+1}(\omega)\}$$

is measurable. Since no jumps occur between time J_n and time t and since X_{post} is a jump process, we have that the above set is equal to

$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_n(\omega) \leq t < J_{n+1}(\omega)\}.$$

We also find

$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_n(\omega) > t) = \cup_{i=0}^{n-1} \{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_i(\omega) \leq t < J_{i+1}(\omega)\}$$

and then

$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_\infty(\omega) > t\} = \lim_{n \to \infty} \{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_n(\omega) > t\}$$

is also measurable. We then find

$$\{\omega \mid J_\infty(\omega) > t\} = \cup_{x \in S_\perp} \{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_\infty(\omega) > t\}.$$

Recall that we assumed that after J_∞ no jumps occur and any trajectory occupies the state $\perp$ indefinitely. We then have

$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x\} = \begin{cases} \{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_\infty(\omega) > t\}, & \text{if } x \neq \perp, \\ \{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x, J_\infty(\omega) > t\} \cup \{\omega \mid J_\infty(\omega) \leq t\}, & \text{if } x = \perp. \end{cases}$$

□

A.1.2 Proof of Proposition 19

For any time-point $t \in \mathbb{R}_{\geq 0}$ we have

$$\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \mid J_\infty > t) = 1.$$

Proof. We will show that for any $t \in \mathbb{R}_{\geq 0}$

$$\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_\infty > t) = \Pr(J_\infty > t).$$

We have

$$\begin{aligned}
&\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_\infty > t) \\
&= \sum_{i=0}^{\infty} \Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_i \leq t < J_{i+1}) \\
&= \sum_{i=0}^{\infty} \Pr(X^{(J_i)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_i \leq t < J_{i+1}),
\end{aligned}$$

since X_{post} is necessarily constant in between jumps. Now, from (6.4), we know that, the probability that $X^{(J_i)}_{\mathsf{post}}$ occupies a state x is non-zero only if there is any reach-trace $\langle w, x\rangle$ to state x from any other state in P. However, the maximal progress assumption (see Definition 29) gives us that such reach-traces only exist for stable states (including $\bot$). We then have

$$\begin{aligned}
&\sum_{i=0}^{\infty} \Pr(X^{(J_i)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_i \leq t < J_{i+1}) \\
&= \sum_{i=0}^{\infty} \Pr(X^{(J_i)}_{\mathsf{post}} \in S_\bot \wedge J_i \leq t < J_{i+1}) \\
&= \sum_{i=0}^{\infty} \Pr(J_i \leq t < J_{i+1}) \\
&= \Pr(J_\infty > t).
\end{aligned}$$

Now we have

$$\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \mid J_\infty > t) = \frac{\Pr(X^{(t)}_{\mathsf{post}} \in S_s \cup \{\bot\} \wedge J_\infty > t)}{\Pr(J_\infty > t)} = 1.$$

□

A.1.3 Proof of Lemma 16

Given a jump-index $n \in \mathbb{N}_0$, a state $x \in S_\perp$, a finite timed path $\sigma \in Paths^{(n)}_{S,A}$, such that $last(\sigma) = x$, and any time-point $t \in \mathbb{R}_{\geq 0}$, we have

$$\Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) = \begin{cases} e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds}, & \text{if } \sigma_t(n) < t \\ 1, & \text{otherwise.} \end{cases}$$

Recall that $\sigma_t(n)$ is the n-th jump-time of σ.

Proof. We first consider time-points $t > \sigma_t(n)$. Consider a time-length $h > 0$. We can then derive a forward equation following Subsection 3.1.3.

$$\begin{aligned} &\Pr(J_{n+1} > t + h \mid Z^{(J_n)} = \sigma) \\ &= \Pr(J_{n+1} > t + h, J_{n+1} > t \mid Z^{(J_n)} = \sigma) \\ &= \Pr(J_1^{(t)} > t + h \mid Z^{(t)} = \sigma) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma). \end{aligned}$$

Now, for the first probability we find that the probability that no jump occurs is equal to the probability that no *Markovian* jump occurs minus the probability that a non-Markovian (i.e., external) jump occurs. We then find that the above equals

$$\begin{aligned} &(\Pr(J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid Z^{(t)} = \sigma) \\ &\qquad - \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid Z^{(t)} = \sigma)) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) \\ &= (\Pr(J_1^{(t)} > t+h \vee X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid X_{\mathsf{post}}^{(t)} = x) \\ &\qquad - \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid Z^{(t)} = \sigma)) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) \\ &= (1 - q_x h - \eta_\sigma^{(t)} h) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) + o(h). \end{aligned}$$

It follows that

$$\frac{d}{dt} \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) = -(q_x + \eta_\sigma^{(t)}) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma).$$

Given the fact that $\Pr(J_{n+1} > t_n \mid Z^{(J_n)} = \sigma) = 1$ we have that

$$\Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) = e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds}$$

is the unique solution to the above differential equation.

For time-points $t \leq \sigma_t(n)$ we have that J_{n+1} must be greater than $\sigma_t(n)$ and then J_{n+1} must also be greater than t with probability one. □

A.1.4 Proof of Theorem 35

Given a behaviour X (with history process Z) of I/O-IMC P with interactive jump scheduler γ and external jump scheduler η, we find for states $x, y \in S_\perp$ and a sequence of actions $w \in \mathcal{L}^V$, that

$$\Pr(Z^{(J_0)} \in \{(x, w, y)\}) = \alpha_x \gamma^{(0)}_{\epsilon,x}(w, y) \tag{A.1}$$

and for a measurable set of timed paths of length $n \in \mathbb{N}$, $H_n \in Paths^{(n)}_{S,A}$, states $y, z \in S_\perp$, and a sequence of actions $w \in \mathcal{L}^V$, we find that

$$\begin{aligned}
&\Pr(Z^{(J_{n+1})} \in H_n \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\
&\quad = \int_{t_1}^{t_2} \left(\int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{t_n}^{t} (q_x + \eta^{(s)}_\sigma) ds} q_{x,y} \gamma^{(t)}_{\sigma,y}(w, z) \right. \\
&\quad \left. + \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{t_n}^{t} (q_y + \eta^{(s)}_\sigma) ds} \eta^{(t)}_\sigma \gamma^{(t)}_{\sigma,y}(w, z) \right) dt.
\end{aligned} \tag{A.2}$$

Recall that $\sigma_z(n)$ is the last state of σ, since it has length n.

Proof. For states $x, y \in S_\perp$ and a sequence of actions $w \in \mathcal{L}^V$ we have

$$\begin{aligned}
&\Pr(Z^{(J_0)} \in \{(x, w, y)\}) \\
&= \Pr(X^{(J_0)}_{\mathsf{post}} = y, W^{(J_0)} = w \mid X^{(J_0)}_{\mathsf{pre}} = x) \Pr(X^{(J_0)}_{\mathsf{pre}} = x).
\end{aligned}$$

We then substitute (6.3) and (6.12) to find (A.1).

For a measurable set of timed paths of length $n \in \mathbb{N}$, $H_n \in Paths^{(n)}_{S,A}$, states $y, z \in S_\perp$, and a sequence of actions $w \in \mathcal{L}^V$, we find that

$$\begin{aligned}
&\Pr(Z^{(J_{n+1})} \in H_n \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}) \\
&= \Pr(X^{(J_{n+1})} = (y, w, z), t_1 < J_{n+1} \leq t_2, Z^{(J_n)} \in H_n) \\
&= \int_{t_1}^{t_2} \Pr(X^{(J_{n+1})} = (y, w, z), t < J_{n+1} \leq t + dt, Z^{(J_n)} \in H_n),
\end{aligned}$$

where we take the *Riemann* integral. Now we have that the above equals

$$\int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(X^{(J_{n+1})} = (y, w, z), t < J_{n+1} \leq t + dt \mid Z^{(J_n)} = \sigma) \Pr(Z^{(J_n)} \in d\sigma), \tag{A.3}$$

where this time we consider the *Lebesgue* integral. That is, for a constant value $c \in [0, 1]$ we find the set of paths $d\sigma$ such that for all $\sigma \in d\sigma$ we have

$$\Pr(X^{(J_{n+1})} = (y, w, z), t < J_{n+1} \leq t + dt \mid Z^{(J_n)} = \sigma) = c.$$

We restrict the integral to paths whose last jump-time lies before time t, since otherwise the above probability must be zero. We can further split the first factor in (A.3) to find that it equals

$$\int_{t_1}^{t_2} \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(X^{(J_1^{(t)})} = (y, w, z), J_1^{(t)} \leq t + dt \mid Z^{(t)} = \sigma) \\ \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) \Pr(Z^{(J_n)} \in d\sigma),$$

and again we split the first probability to consider the Markovian and interactive part of the jump separately to find

$$\begin{aligned} \int_{t_1}^{t_2} \Bigg(& \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \\ & \Pr(X_{\mathsf{post}}^{(J_{n+1})} = z, W^{(J_{n+1})} = w \mid X_{\mathsf{pre}}^{(J_{n+1})} = y, J_{n+1} = t, Z^{(J_n)} = \sigma) \\ & \cdot \Pr(X_{\mathsf{pre}}^{(J_1^{(t)})} = y, J_1^{(t)} \leq t + dt \mid X_{\mathsf{post}}^{(t)} = x) \\ & \quad \cdot \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) \Pr(Z^{(J_n)} \in d\sigma) \\ + & \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} \Pr(X_{\mathsf{post}}^{(J_{n+1})} = z, W^{(J_{n+1})} = w \mid X_{\mathsf{pre}}^{(J_{n+1})} = y, J_{n+1} = t, Z^{(J_n)} = \sigma) \\ & \cdot \Pr(X_{\mathsf{pre}}^{(J_1^{(t)})} = y, J_1^{(t)} \leq t + dt \mid Z^{(t)} = \sigma) \\ & \quad \cdot \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma) \Pr(Z^{(J_n)} \in d\sigma) \Bigg), \end{aligned}$$

where we applied (6.7) to the second probability (i.e., the Markovian jump probability from x to y). Note that we must distinguish between the case that the last state of σ is different from y, which means the $n+1$-th jump is Markovian (or combined), and the case that the last state of σ is the same as y, which means that the $n+1$-th jump is an external interactive jump. Now, we substitute (6.12), (6.14), (6.5), and (6.15) into the above to find

$$\int_{t_1}^{t_2} \Bigg(\int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = x \neq y}} \gamma_{\sigma,y,w,z}^{(t)} q_{x,y} dt e^{-\int_{\sigma_t(n)}^{t} (q_x + \eta_\sigma^{(s)}) ds} \Pr(Z^{(J_n)} \in d\sigma) \\ + \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ \sigma_z(n) = y}} \gamma_{\sigma,y,w,z}^{(t)} \eta_\sigma^{(t)} dt e^{-\int_{\sigma_t(n)}^{t} (q_y + \eta_\sigma^{(s)}) ds} \Pr(Z^{(J_n)} \in d\sigma) \Bigg).$$

Note that we use the fact that, since we take the Riemann integral to range over time, the interval dt converges to zero, which means the term $o(dt)$ vanishes in (6.5) and (6.14). We can now rearrange the above to find (A.2). □

A.1.5 Proof of Theorem 36

Given a measurable set H_{n-1} of paths of length $n-1$, let $t_1 < t_2$ be two time-points, let $y, z \in S_\perp$ be two states, and let $w \in \mathcal{L}^V$ be a sequence of visible actions. For the measurable set of paths

$$H_n = H_{n-1} \times (t_1, t_2] \times \{y\} \times \{w\} \times \{z\}.$$

we find

$$\Pr(Z^{(t)} \in H_n) = \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t \\ x = \sigma_z(n)}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_x + \eta_\sigma^{(s)}) ds}. \tag{A.4}$$

Proof. First of all, we will compute the probability that X follows a path in H_n up to the n-th jump and then remains in state z perpetually, i.e., $J_{n+1} = \infty$. We have

$$\begin{aligned}
&\Pr(Z^{(J_n)} \in H_n, J_{n+1} = \infty) \\
&= \Pr(Z^{(J_n)} \in H_n) - \Pr(Z^{(J_{n+1})} \in H_n, J_{n+1} \in \mathbb{R}_{\geq 0}) \\
&= \Pr(Z^{(J_n)} \in H_n) - \Pr(Z^{(J_{n+1})} \in H_n \times \mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp),
\end{aligned}$$

by the law of total probability. We now substitute (6.17) to find that

$$\begin{aligned}
&\Pr(Z^{(J_{n+1})} \in H_n \times \mathbb{R}_{\geq 0} \times S_\perp \times \mathcal{L}^V \times S_\perp) \\
&= \sum_{y' \in S_\perp, y' \neq z} \sum_{w' \in \mathcal{L}^V} \sum_{z \in S_\perp} \\
&\int_0^\infty \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{t_n}^t (q_z + \eta_\sigma^{(s)}) ds} q_{z,y'} \gamma_{\sigma,y'}^{(t)}(w', z') dt \\
&+ \sum_{w' \in \mathcal{L}^V} \sum_{z \in S_\perp} \\
&\int_0^\infty \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{t_n}^t (q_z + \eta_\sigma^{(s)}) ds} \eta_\sigma^{(t)} \gamma_{\sigma,z}^{(t)}(w', z') dt,
\end{aligned}$$

where we used the fact that all paths in H_n end with the state z. Since $\gamma_{\sigma,y'}^{(t)}$ is a probability function we have that $\sum_{w' \in \mathcal{L}^V} \sum_{z' \in S_\perp} \gamma_{\sigma,y'}^{(t)} = 1$. Moreover, we have $\sum_{y' \in S_\perp, y' \neq z} q_{z,y'} = q_z$ and then it follows that the above equals

$$\begin{aligned}
&\int_0^\infty \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_z + \eta_\sigma^{(s)}) ds} q_z dt \\
&+ \int_0^\infty \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_z + \eta_\sigma^{(s)}) ds} \eta_\sigma^{(t)} dt \\
&= \int_0^\infty \int_{\substack{\sigma \in H_n \\ \sigma_t(n) < t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_z + \eta_\sigma^{(s)}) ds} (q_z + \eta_\sigma^{(t)}) dt \\
&= \int_{\sigma \in H_n} \Pr(Z^{(J_n)} \in d\sigma) \int_{\sigma_t(n)}^\infty e^{-\int_{\sigma_t(n)}^t (q_z + \eta_\sigma^{(s)}) ds} (q_z + \eta_\sigma^{(t)}) dt,
\end{aligned}$$

by Fubini's theorem. Now we can solve the inner integral to find

$$\int_{\sigma\in H_n} \Pr(Z^{(J_n)} \in d\sigma) \left[-e^{-\int_{\sigma_t(n)}^{t}(q_z+\eta_\sigma^{(s)})ds}\right]_{t=\sigma_t(n)}^{t=\infty}$$
$$= \int_{\sigma\in H_n} \Pr(Z^{(J_n)} \in d\sigma)(1 - e^{-\int_{\sigma_t(n)}^{\infty}(q_z+\eta_\sigma^{(s)})ds}).$$

For the exponential distribution we find

$$e^{-\int_{\sigma_t(n)}^{\infty}(q_z+\eta_\sigma^{(s)})ds} = \begin{cases} 1, & \text{if } q_z > 0, \\ e^{-\int_{\sigma_t(n)}^{\infty}\eta_\sigma^{(s)}ds}. & \text{if } q_z = 0, \end{cases}$$

and then we find

$$\Pr(Z^{(J_n)} \in H_n, J_{n+1} = \infty)$$
$$= \begin{cases} 0, & \text{if } q_z > 0, \\ \int_{\sigma\in H_n} \Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^{\infty}\eta_\sigma^{(s)}ds}, & \text{if } q_z = 0. \end{cases} \tag{A.5}$$

Now we consider the probability that X follows a path in H_n and then does not jump before time $t \in \mathbb{R}_{\geq 0}$, i.e., the history process Z is in H_n at time t. We have,

$$\Pr(Z^{(t)} \in H_n) = \int_{\substack{\sigma\in H_n \\ \sigma_t(n)<t}} \Pr(Z^{(J_n)} \in d\sigma, J_{n+1} > t)$$
$$= \int_{\substack{\sigma\in H_n \\ \sigma_t(n)<t}} \Pr(Z^{(J_n)} \in d\sigma) \Pr(J_{n+1} > t \mid Z^{(J_n)} = \sigma).$$

Now, we apply (6.15) to find (A.4). □

A.1.6 Proof of Lemma 17

Given distinct states $x, y \in S_\perp$, a path-length $n \in \mathbb{N}_0$, time-points $t < t + h \in \mathbb{R}_{\geq 0}$, and a measurable set of times paths H_n of length n, such that for each path σ in H_n we have $\sigma_t(n) < t$ and $\sigma_z(n) = x$ and $\Pr(Z^{(t)} \in H_n) > 0$, we have

$$\Pr(X_{\mathsf{pre}}^{(J_n+1)} = y, J_{n+1} \leq t + h \mid Z^{(t)} \in H_n) = q_{x,y}h + o(h). \tag{A.6}$$

Proof. We have

$$\Pr(X_{\mathsf{pre}}^{(J_n+1)} = y, J_{n+1} \leq t + h \mid Z^{(t)} \in H_n)$$
$$= \int_{\sigma\in H_n} \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n) \frac{\Pr(X_{\mathsf{pre}}^{(J_{n+1})} = y, J_{n+1} \leq t + h, Z^{(t)} \in d\sigma)}{\Pr(Z^{(t)} \in d\sigma)}$$
$$= \int_{\sigma\in H_n} \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n)$$
$$\cdot \frac{\Pr(Z^{(J_n+1)} \in d\sigma \times (t, t+h] \times \{y\} \times \mathcal{L}^V \times S_\perp)}{\Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^{t}(q_x+\eta_\sigma^{(s)})ds}}, \tag{A.7}$$

where we applied (6.18). We now apply (A.2) to the numerator, keeping in mind that for each path in H_n we have that its last state is x which is unequal to y.

$$\begin{aligned}
&\Pr(Z^{(J_n+1)} \in d\sigma \times (t, t+h] \times \{y\} \times \mathcal{L}^V \times S_\perp) \\
&= \sum_{w \in \mathcal{L}^V} \sum_{z \in S_\perp} \int_t^{t+h} \Pr(Z^{(J_n)} \in d\sigma) q_{x,y} \gamma^{(u)}_{\sigma,y}(w,z) e^{-\int_{\sigma_t(n)}^u (q_x + \eta^{(s)}_\sigma) ds} du \\
&= \Pr(Z^{(J_n)} \in d\sigma) \int_t^{t+h} q_{x,y} e^{-\int_{\sigma_t(n)}^u (q_x + \eta^{(s)}_\sigma) ds} \sum_{w \in \mathcal{L}^V} \sum_{z \in S_\perp} \gamma^{(u)}_{\sigma,y}(w,z) du \\
&= \Pr(Z^{(J_n)} \in d\sigma) \int_t^{t+h} q_{x,y} e^{-\int_{\sigma_t(n)}^u (q_x + \eta^{(s)}_\sigma) ds} du,
\end{aligned}$$

since $\gamma^{(u)}_{\sigma,y}$ is a probability function and must sum up to one. We now apply the Taylor expansion around $h = 0$ to find that the above equals

$$\Pr(Z^{(J_n)} \in d\sigma) q_{x,y} e^{-\int_{\sigma_t(n)}^t (q_x + \eta^{(s)}_\sigma) ds} h + o(h). \tag{A.8}$$

Now we can substitute (A.8) into (A.7) to find

$$\begin{aligned}
&\int_{\sigma \in H_n} \frac{\Pr(Z^{(J_n)} \in d\sigma) q_{x,y} e^{-\int_{\sigma_t(n)}^t (q_x + \eta^{(s)}_\sigma) ds} h + o(h)}{\Pr(Z^{(J_n)} \in d\sigma) e^{-\int_{\sigma_t(n)}^t (q_x + \eta^{(s)}_\sigma) ds}} \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n) \\
&= \int_{\sigma \in H_n} q_{x,y} h \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n) + o(h) \\
&= q_{x,y} h \int_{\sigma \in H_n} \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n) + o(h),
\end{aligned}$$

and since $\int_{\sigma \in H_n} \Pr(Z^{(t)} \in d\sigma \mid Z^{(t)} \in H_n)$ equals one we have that (A.6) holds. □

A.1.7 Proof of Theorem 37

Given an I/O-IMC P, an interactive jump scheduler γ for P, and an external jump scheduler η for P, we have that the interactive jump process X with probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$, where $\mathcal{P}$ is constructed as per (6.22) and (6.23), is a behaviour of P.

Proof. We show that the four requirements for being a behaviour of P hold for X.

Requirement (6.3). For the sake of simplicity, we will use the function f defined in (6.20) instead of the function $\mathcal{P}$. For any state $x \in S_\perp$ we have

$$\begin{aligned}
\Pr(X^{(0)}_{\mathsf{pre}} = x) &= \Pr(Z^{(0)} \in \{x\} \times \mathcal{L}^V \times S_\perp) \\
&= \sum_{w \in \mathcal{L}^V} \sum_{y \in S_\perp} f_0(\{(x,w,y)\}) \\
&= \alpha_x \sum_{w \in \mathcal{L}^V} \sum_{y \in S_\perp} \gamma^{(0)}_{\epsilon,x}(w,y).
\end{aligned}$$

Since $\gamma^{(0)}_{\epsilon,x}$ is a probability function and must sum up to one, we have

$$\Pr(X^{(0)}_{\mathsf{pre}} = x) = \alpha_x,$$

i.e., (6.3) holds.

Requirement (6.4) It follows directly from the definition of the interactive jump scheduler that interactive jumps are only assigned positive probability if there is an appropriate fair reach-trace.

Requirement (6.5) Consider distinct states $x, y \in S_\perp$ and time-points $t < t + h \in \mathbb{R}_{\geq 0}$ such that $\Pr(X^{(t)}_{\mathsf{post}} = x) > 0$. We first look at the case $x \neq \perp$. We then have, by construction, that $X^{(t)}_{\mathsf{post}} = x$ implies that t is smaller than the explosion time J_∞, since we have assumed that each path occupies the state $\perp$ for any time-point greater than or equal to J_∞. We can then apply the law of total probability to fix the number of jumps before time t to find

$$\begin{aligned}
&\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x) \\
&= \sum_{n=0}^{\infty} \Pr(J_{n+1} \leq t + h, X^{(J_{n+1})}_{\mathsf{pre}} = y, J_n \leq t < J_{n+1} \mid X^{(t)}_{\mathsf{post}} = x) \\
&= \sum_{n=0}^{\infty} \Pr(J_{n+1} \leq t + h, X^{(J_{n+1})}_{\mathsf{pre}} = y \mid J_n \leq t < J_{n+1}, X^{(t)}_{\mathsf{post}} = x) \\
&\qquad \Pr(J_n \leq t < J_{n+1} \mid X^{(t)}_{\mathsf{post}} = x).
\end{aligned}$$

In terms of the history process we find for the first probability that, given some $n \in \mathbb{N}_0$,

$$\begin{aligned}
&\Pr(J_{n+1} \leq t + h, X^{(J_{n+1})}_{\mathsf{pre}} = y \mid J_n \leq t < J_{n+1}, X^{(t)}_{\mathsf{post}} = x) \\
&= \Pr(J_{n+1} \leq t+h, X^{(J_{n+1})}_{\mathsf{pre}} = y \mid Z^{(t)} \in (S_\perp \times \mathcal{L}^V \times S_\perp \times \mathbb{R}_{\geq 0})^n \times S_\perp \times \mathcal{L}^V \times \{x\}) \\
&= q_{x,y} h + o(h),
\end{aligned}$$

due to (6.24). It then follows that

$$\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x) = q_{x,y} h + o(h),$$

i.e., (6.4) holds for the case $x \neq \perp$.

For the case $x = \perp$, we have that no jumps will occur (again by construction). This means that the Markovian jump probability is zero, which is $o(h)$, i.e.,

$$\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = \perp) = o(h).$$

This matches the fact that $q_{\perp,y} = 0$ for all y.

Requirement (6.6) Since the probability of a Markovian jump and the probability of an external jump in a time-interval $(t, t + h]$ are proportionate to h, it follows that

the probability of two jumps in such a time-interval is proportionate to h^2 and then this probability is $o(h)$.

Requirement (6.7) Adding extra history conditions to the Markovian jump probabilities does not affect the applicability of (6.24) since it holds for all histories. It immediately follows that the Markovian jump probabilities are indeed memoryless up to $o(h)$. □

A.1.8 Proof of Proposition 21

Given an interactive jump process $\tilde{X}$ for the I/O-IMC $\tilde{P} = P \| \bar{P}$ defined on a probability space that satisfies Proposition 18, we find that the following probabilities for the projected interactive jump process $X = \tilde{X} \downarrow P$ are measurable.

1. For any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $w_i \in \mathcal{L}^V$, the set of trajectories where the i-th interactive jump starts in x_i, ends in y_i and has sequence w_i,
$$\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\},$$
is measurable.

2. For any time-points $t, h \in \mathbb{R}_{\geq 0}$ we have, that the set of trajectories where the first jump after time t occurs before time $t + h$,
$$\{\omega \mid J_1^{(t)}(\omega) \leq t + h\},$$
is measurable.

3. For any time-point $t \in \mathbb{R}_{\geq 0}$ and any state $x \in S_\perp$ we have, that the set of trajectories where the stochastic process X_{post} occupies state x at time t,
$$\{\omega \mid X_{\mathsf{post}}^{(t)}(\omega) = x\},$$
is measurable.

Proof. We will prove Proposition 21 by assume the events for X are indeed measurable and then expressing them in terms of measurable events for $\tilde{X}$.

We have seen in Proposition 20 that every jump of X corresponds to a jump of $\tilde{X}$. We then have, for any jump-index i, states $x_i, y_i \in S_\perp$, and sequence $w_i \in \mathcal{L}^V$, that,

$$\begin{aligned}&\{\omega \mid X^{(J_i)}(\omega) = (x_i, w_i, y_i)\}\\ &= \cup_{\bar{x},\bar{y} \in \bar{S}_\perp} \cup_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V : \\ \tilde{w} \downarrow P = w_i}} \cup_{j=0}^{\infty} \{\omega \mid \tilde{X}^{(\tilde{J}_j)}(\omega) = (x_i \| \bar{x}, \tilde{w}, y_i \| \bar{y}), \tilde{J}_j(\omega) = J_i(\omega)\}.\end{aligned}$$

It then remains to show that the event $\{\tilde{J}_j(\omega) = J_i(\omega)\}$ is measurable. For $i = 0$ we have $\tilde{J}_0 = J_0$ by definition. We now show that the event is measurable for $i = 1$. First of all, we note that $J_1 > \tilde{J}_0$. We consider an index $j \in \mathbb{N}$. In order for $\tilde{J}_j$ to be equal to

J_1 we have that each jump before $\tilde{J}_{j-1}$ does not register as a jump of X and jump $\tilde{J}_j$ does. We then have, for $j \in \mathbb{N}_0$,

$$\begin{aligned}
&\{\omega \mid \tilde{J}_j(\omega) = J_1(\omega)\} \\
&= \{\omega \mid X_{\mathsf{post}}^{(\tilde{J}_j)}(\omega) \neq X_{\mathsf{pre}}^{(\tilde{J}_j)}(\omega) \vee \tilde{W}^{(\tilde{J}_j)}(\omega)\!\downarrow\! P \neq \epsilon \vee X_{\mathsf{pre}}^{(\tilde{J}_j)}(\omega) \neq X_{\mathsf{post}}^{(\tilde{J}_{j-1})}(\omega), \\
&\quad \forall 1 \leq k < j \cdot \neg(X_{\mathsf{post}}^{(\tilde{J}_k)}(\omega) \neq X_{\mathsf{pre}}^{(\tilde{J}_k)}(\omega) \vee \tilde{W}^{(\tilde{J}_k)}(\omega)\!\downarrow\! P \neq \epsilon \vee X_{\mathsf{pre}}^{(\tilde{J}_k)}(\omega) \neq X_{\mathsf{post}}^{(\tilde{J}_{k-1})}(\omega)\},
\end{aligned}$$

where we simply applied the definition of a jump-time for X. Note that the above events are all measurable by enumerating all possible states and action-sequences for $\tilde{X}$.

We now consider the event

$$\{\omega \mid J_1^{(t)}(\omega) \leq t + h\},$$

for some time-points $t, t + h \in \mathbb{R}_{\geq 0}$. Again, we must consider which jump of $\tilde{X}$ corresponds to the first jump of X after time t. We then find that the above equals

$$\cup_{j=1}^{\infty}\{\omega \mid \tilde{J}_j^{(t)}(\omega) \leq t + h, \tilde{J}_j^{(t)}(\omega) = J_1^{(t)}(\omega)\}.$$

Now the question remains whether the events $\{\omega \mid \tilde{J}_j^{(t)}(\omega) \leq t + h\}$ are measurable for $j > 1$. To do this we must fix the time-points of the preceding jumps. For instance, for $j = 2$ we find

$$\{\omega \mid \tilde{J}_2^{(t)}(\omega) \leq t + h)\} = \{\omega \mid \tilde{J}_1^{(t_1+dt1)}(\omega) \leq t + h, t_1 \leq \tilde{J}_1^{(t)}(\omega) \leq t_1 + dt_1\}$$

and similarly we find that this event is also measurable for all $j > 2$.

Finally, we turn to the event where X_{post} occupies a state $x \in S_\perp$ at time $t \in \mathbb{R}_{\geq 0}$. But we simply find

$$\{\omega \mid X_{\mathsf{post}}^{(t)} = x(\omega)\} = \cup_{\bar{x} \in \bar{S}_\perp}\{\omega \mid \tilde{X}_{\mathsf{post}}^{(t)}(\omega) = x \| \bar{x}\}.$$

□

A.1.9 Proof of Theorem 38

Given a behaviour $\tilde{X}^{(t)}, t \in \mathbb{R}_{\geq 0}$ of $\tilde{P}$, its projections onto P and $\bar{P}$ are compatible behaviours of P and $\bar{P}$ respectively.

Proof. Let X and $\bar{X}$ be the projected behaviours $\tilde{X}\downarrow P$ and $\tilde{X}\downarrow \bar{P}$ respectively. We will first show that X is indeed a behaviour of P by showing that it satisfies all the requirements of Definition 73.

Requirement (6.3): initial distribution. For the initial distribution of X we find by the law of total probability that for any state $x \in S$ we have,

$$\begin{aligned}
\Pr(X_{\mathsf{pre}}^{(0)} = x) &= \sum_{\bar{x} \in \bar{S}} \Pr(\tilde{X}_{\mathsf{pre}}^{(0)} = x \| \bar{x}) \\
&= \sum_{\bar{x} \in \bar{S}} \tilde{\alpha}(x \| \bar{x}).
\end{aligned}$$

From Definition of 52 we know that the initial probabilities of $\tilde{X}$ are derived from the initial distributions α and $\bar{\alpha}$, such that the above equals

$$\begin{aligned}&\sum_{\bar{x}\in\bar{S}}\alpha(x)\bar{\alpha}(\bar{x})\\&=\alpha(x)\sum_{\bar{x}\in\bar{S}}\bar{\alpha}(\bar{x})\\&=\alpha(x).\end{aligned}$$

Recall that the probability to start in the state $\perp$ is always zero.

Requirement (6.4): interactive jumps. We must now show that any interactive jump in X occurs with probability greater than zero only when there is an appropriate reach-trace in P. Consider now an interactive jump from state $y \in S_\perp$ to state $z \in S_\perp$ with sequence $w \in \mathcal{L}^V$ and let this be the i-th jump of X. We then find, by the law of total probability that,

$$\begin{aligned}&\Pr(X^{(J_i)}_{\mathsf{post}} = z, W^{(J_i)} = w \mid X^{(J_i)}_{\mathsf{pre}} = y)\\&= \sum_{\bar{y}\in\bar{S}_\perp}\sum_{\bar{z}\in\bar{S}_\perp}\sum_{\substack{\tilde{w}\in\tilde{A}^*:\\ \tilde{w}\downarrow P=w}} \Pr(\tilde{X}^{(t)}_{\mathsf{post}} = z\|\bar{z}, \tilde{W}^{(t)} = \tilde{w} \mid \tilde{X}^{(t)}_{\mathsf{pre}} = y\|\bar{y})\\&\qquad\qquad \cdot \Pr(\tilde{X}^{(t)}_{\mathsf{pre}} = y\|\bar{y} \mid X^{(t)}_{\mathsf{pre}} = y)\end{aligned}$$

Recall that, since W is the projection of $\tilde{W}$ onto A the probability $\Pr(\tilde{W}^{(J_i)} = \tilde{w}, W^{(J_i)} = w)$ is zero unless $\tilde{w}\downarrow P = w$.

We now have that if

$$\Pr(X^{(t)}_{\mathsf{post}} = z, W^{(t)} = w \mid X^{(t)}_{\mathsf{pre}} = y) > 0$$

then there is some combination $\bar{y}, \bar{z}, \tilde{w}$ such that

$$\Pr(\tilde{X}^{(t)}_{\mathsf{post}} = z\|\bar{z}, \tilde{W}^{(t)} = \tilde{w} \mid \tilde{X}^{(t)}_{\mathsf{pre}} = y\|\bar{y}) > 0.$$

Because $\tilde{X}$ is a behaviour of $\tilde{P}$ this means $(\tilde{w}, z\|\bar{z}) \in \mathit{FairRT}(\mathit{IOA}(y\|\bar{y}))$. By Proposition 13, we have that

$$\mathit{IOA}(y\|\bar{y}) = \mathit{IOA}(y)\|\mathit{IOA}(\bar{y})$$

and then Corollary 7 gives us that (w, z), the projection of reach-trace $(\tilde{w}, z\|\bar{z})$ onto P is indeed a fair reach-trace of $\mathit{IOA}(y)$.

Requirement (6.5): Markovian jump. We now show that the probability of a Markovian jump from a state $x \in S_\perp$ to a distinct state $y \in S_\perp$ equals $q_{x,y}h + o(h)$. By the law of total probability we have

$$\begin{aligned}&\Pr(J^{(t)}_1 \le t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{post}} = x)\\&= \sum_{\bar{x}\in\bar{S}_\perp} \Pr(J^{(t)}_1 \le t+h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid \tilde{X}^{(t)}_{\mathsf{post}} = x\|\bar{x}) \Pr(\tilde{X}^{(t)}_{\mathsf{post}} = x\|\bar{x} \mid X^{(t)}_{\mathsf{post}} = x).\end{aligned}$$

Since any jump of X implies a jump of $\tilde{X}$ we have that the jump-time $\tilde{J}_1^{(t)}$ is either equal to $J_1^{(t)}$ or less than $J_1^{(t)}$. However, the case that $\tilde{J}_1^{(t)} < J_1^{(t)} \leq t+h$ implies that two distinct jumps occur within the time-interval $[t, t+h]$, but (6.6) tells us this occurs with probability $o(h)$. We then find that the Markovian jump probability of X equals

$$\sum_{\bar{x}\in\bar{S}_\perp}\sum_{\bar{y}\in\bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{y} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}) \cdot \Pr(\tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x} \mid X_{\mathsf{post}}^{(t)} = x) + o(h).$$

Since $x \neq y$ we can apply (5.3) to find that the first factor equals $o(h)$ if $\bar{x} \neq \bar{y}$. The above then equals

$$\begin{aligned}
&\sum_{\bar{x}\in\bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{x} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}) \\
&\qquad \cdot \Pr(\tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x} \mid X_{\mathsf{post}}^{(t)} = x) + o(h) \\
&= (q_{x,y}h + o(h)) \sum_{\bar{x}\in\bar{S}_\perp} \Pr(\tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x} \mid X_{\mathsf{post}}^{(t)} = x) + o(h) \\
&= q_{x,y}h + o(h).
\end{aligned}$$

Requirement (6.6): Two jumps. Whenever X makes two jumps in a time-interval $[t, t+h]$, with $t \in \mathbb{R}_{\geq 0}$, $h > 0$, we have that $\tilde{X}$ also makes at least two jumps in this time-interval. It immediately follows that the probability that X makes two jumps in $[t, t+h]$ is $o(h)$.

Requirement (6.7): Local Markov property. For the sake of simplicity, we will prove the "Markov property up to $o(h)$" for a single extra condition in the past. The general property follows in a similar way. For two distinct states $x, y \in S_\perp$, a time-point $t \in \mathbb{R}_{\geq 0}$, a time-length $h > 0$, as well as a state $x_1 \in S_\perp$ and a time-point $t_1 < t$ we find

$$\begin{aligned}
&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x, X_{\mathsf{post}}^{(t_1)} = x_1) \\
&= \sum_{\bar{x}\in\bar{S}_\perp}\sum_{\bar{x}_1\in\bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{x} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}, \tilde{X}_{\mathsf{post}}^{(t_1)} = x_1\|\bar{x}_1) \\
&\qquad \cdot \Pr(\bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, \bar{X}_{\mathsf{post}}^{(t_1)} = \bar{x}_1 \mid X_{\mathsf{post}}^{(t)} = x, X_{\mathsf{post}}^{(t_1)} = x_1) + o(h) \\
&= (q_{x,y}h + o(h)) \\
&\qquad \cdot \sum_{\bar{x}\in\bar{S}_\perp}\sum_{\bar{x}_1\in\bar{S}_\perp} \Pr(\bar{X}_{\mathsf{post}}^{(t)} = \bar{x}, \bar{X}_{\mathsf{post}}^{(t_1)} = \bar{x}_1 \mid X_{\mathsf{post}}^{(t)} = x, X_{\mathsf{post}}^{(t_1)} = x_1) + o(h) \\
&= q_{x,y}h + o(h).
\end{aligned}$$

The last step is due to the Markov property up to $o(h)$ for $\tilde{X}$ and (6.5). Since we have already show that (6.5) holds for X, i.e.,

$$\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x) = q_{x,y}h + o(h),$$

it follows that

$$\begin{aligned}&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x, X_{\mathsf{post}}^{(t_1)} = x_1)\\ &= \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x) + o(h).\end{aligned}$$

Note that we have *not* proven equality since, although both probabilities equal $q_{x,y}h + o(h)$, the two $o(h)$ functions may in fact be different.

We have now shown that X is indeed a behaviour of P. Similarly, we can show that $\bar{X}$ is a behaviour of $\bar{P}$. It now remains to show that these two behaviours are compatible.

Requirement (6.25): independence of initial distributions. For a pair of states $x \in S_\perp$, $\bar{x} \in \bar{S}_\perp$ we find, by the definition of parallel composition and the fact that $\tilde{X}$ satisfies (6.3), that

$$\begin{aligned}&\Pr(X_{\mathsf{pre}}^{(0)} = x, \bar{X}_{\mathsf{pre}}^{(0)} = \bar{x})\\ &= \Pr(\tilde{X}_{\mathsf{pre}}^{(0)} = x \| \bar{x})\\ &= \alpha_x \bar{\alpha}_{\bar{x}}\\ &= \Pr(X_{\mathsf{pre}}^{(0)} = x) \Pr(\bar{X}_{\mathsf{pre}}^{(0)} = \bar{x}).\end{aligned}$$

The last equality follows from the fact that both X and $\bar{X}$ satisfy (6.3).

Requirement (6.26): synchronization of traces. This follows directly from the fact that W and $\bar{W}$ are projections of $\tilde{W}$.

Requirement (6.27): synchronization of time-divergence. This follows directly from the fact that X_{pre} and $\bar{X}_{\mathsf{pre}}$ are projections of $\tilde{X}_{\mathsf{pre}}$ and the fact that X_{post} and $\bar{X}_{\mathsf{post}}$ are projections of $\tilde{X}_{\mathsf{post}}$.

Requirement (6.28): Two jumps. This follows directly from the fact that for $\tilde{X}$, it holds that two jumps occur in a time-interval $[t, t+h], t \in \mathbb{R}_{\geq 0}, h > 0$, with probability $o(h)$.

For the requirements (6.29), (6.30), and (6.31), that the Markovian jumps of X and $\bar{X}$ are independent up to $o(h)$, we will first show that these conditions hold without any additional history conditions. The general conditions then immediately follow from the fact that X, $\bar{X}$, and $\tilde{X}$ are all behaviours of I/O-IMCs and satisfy the "Markov property up to $o(h)$" for Markovian jumps, (6.7).

Requirement (6.29): Independence up to $o(h)$ Consider distinct states $x, y \in S_\perp$, a state $\bar{x} \in \bar{S}_\perp$, a time-point $t \in \mathbb{R}_{\geq 0}$, and a time-length $h > 0$. For the Markovian jump probability of X given that $\bar{X}$ occupies state $\bar{x}$ at time t we find

$$\begin{aligned}&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x})\\ &= \sum_{\bar{y} \in \bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y \| \bar{y} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x \| \bar{x}) + o(h)\\ &= \sum_{\bar{y} \in \bar{S}_\perp} \tilde{q}_{x\|\bar{x}, y\|\bar{y}} h + o(h).\end{aligned}$$

Note that the event, that $\tilde{X}$ experiences a jump after time t but before $J_1^{(t)}$, occurs with probability $o(h)$ because of (6.6). From (5.3) we know that

$$\tilde{q}_{x\|\bar{x},y\|\bar{y}} = \begin{cases} q_{x,y}, & \text{if } \bar{x} = \bar{y}, \\ 0, & \text{if } \bar{x} \neq \bar{y}. \end{cases}$$

We then find that the above equals

$$q_{x,y}h + o(h).$$

Now, for the Markovian jump probability of X with no condition on the location of $\bar{X}$ at time t we find (where $\bar{x}$ is now a free variable)

$$\begin{aligned} &\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x) \\ &= \sum_{\bar{x}\in\bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{x} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}) \\ &\qquad \cdot \Pr(\bar{X}_{\mathsf{post}}^{(t)} = \bar{x} \mid X_{\mathsf{post}}^{(t)} = x) + o(h), \\ &= q_{x,y}h \sum_{\bar{x}\in\bar{S}_\perp} \Pr(\bar{X}_{\mathsf{post}}^{(t)} = \bar{x} \mid X_{\mathsf{post}}^{(t)} = x) + o(h) \\ &= q_{x,y}h + o(h). \end{aligned}$$

It then follows that

$$\begin{aligned} &\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}) \\ &= \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x) + o(h). \end{aligned}$$

The general case, with additional history conditions follows by applying the Markov property up to $o(h)$ for $\tilde{X}$ and X.

Requirement (6.30): Independence up to $o(h)$ Symmetric to the proof of (6.29).

Requirement (6.31): Independence up to $o(h)$**.** Consider distinct states $x, y \in S_\perp$, distinct states $\bar{x}, \bar{y} \in \bar{S}_\perp$, time-point $t \in \mathbb{R}_{\geq 0}$, and time-length $h > 0$. We first look at the left-hand side of (6.31), i.e.,

$$\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y, \bar{J}_1^{(t)} \leq t+h, \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{y} \mid X_{\mathsf{post}}^{(t)} = x, \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}).$$

For the case that $J_1^{(t)} \neq \bar{J}_1^{(t)}$ we have that $\tilde{X}$ experiences two jumps in time-interval $[t, t+h]$ which occurs with probability $o(h)$. For the case $J_1^{(t)} = \bar{J}_1^{(t)}$ we have that $\tilde{X}$ makes a Markovian jump to state $y\|\bar{y}$. We then have that the above equals

$$\begin{aligned} &\Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{y} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}) \\ &= \tilde{q}_{x\|\bar{x},y\|\bar{y}}h + o(h) = o(h). \end{aligned}$$

The last equality follows from (5.3). For the right-hand side of (6.31) we use the fact that X and $\bar{X}$ are behaviours of P respectively $\bar{P}$, which means they satisfy (6.5). We then find

$$\begin{aligned}&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{post}}^{(t)} = x)\\ &\quad \cdot \Pr(\bar{J}_1^{(t)} \leq t+h, \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{y} \mid \bar{X}_{\mathsf{post}}^{(t)} = \bar{x}) + o(h)\\ &= (q_{x,y}h + o(h))(\bar{q}_{\bar{x},\bar{y}}h + o(h)) + o(h) = o(h).\end{aligned}$$

We find the same for the right-hand side of (6.31) and then this shows (6.31) holds. □

A.1.10 Proof of Theorem 39

Given a stable interactive jump process $\tilde{X}$ for $P\|\bar{P}$, if the projections of $\tilde{X}$ onto P and $\bar{P}$ are compatible behaviours of P respectively $\bar{P}$, then $\tilde{X}$ is a behaviour of $P\|\bar{P}$.

Proof. Let $X = \tilde{X}\downarrow P$ and $\bar{X} = \tilde{X}\downarrow \bar{P}$ be the projections of $\tilde{X}$ onto P respectively $\bar{P}$. We then have for any time-point t that

$$\begin{aligned}&\tilde{X}_{\mathsf{pre}}^{(t)} = X_{\mathsf{pre}}^{(t)} \| \bar{X}_{\mathsf{post}}^{(t)},\\ &\tilde{W}^{(t)}\downarrow A^V = W^{(t)},\\ &\tilde{W}^{(t)}\downarrow \bar{A}^V = \bar{W}^{(t)}, \text{ and}\\ &\tilde{X}_{\mathsf{post}}^{(t)} = X_{\mathsf{post}}^{(t)} \| \bar{X}_{\mathsf{post}}^{(t)}.\end{aligned}$$

We will use the above and the fact that X and $\bar{X}$ are compatible behaviours of P respectively $\bar{P}$ to show that $\tilde{X}$ is a behaviour of $P\|\bar{P}$.

Requirement (6.3): initial distribution. For states $x \in S_\perp$, $\bar{x} \in \bar{S}_\perp$ we have

$$\begin{aligned}\Pr(\tilde{X}_{\mathsf{pre}}^{(0)} = x\|\bar{x}) &= \Pr(X_{\mathsf{pre}}^{(0)} = x, \bar{X}_{\mathsf{pre}}^{(0)} = \bar{x})\\ &= \Pr(X_{\mathsf{pre}}^{(0)} = x)\Pr(\bar{X}_{\mathsf{pre}}^{(0)} = \bar{x}),\end{aligned}$$

because of (6.25). The above equals $\alpha_x\bar{\alpha}_{\bar{x}}$ which matches the initial distribution of $\tilde{X}$ as required by (6.3).

Requirement (6.4): interactive jumps. Consider states $y, z \in S_\perp$, $\bar{y}, \bar{z} \in \bar{S}_\perp$, and a word $\tilde{w} \in \tilde{\mathcal{L}}^V$ such that

$$\Pr(\tilde{X}_{\mathsf{post}}^{(\tilde{J}_i)} = z\|\bar{z}, \tilde{W}^{(\tilde{J}_i)} = \tilde{w} \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_i)} = y\|\bar{y}) > 0.$$

We must show that it follows that $\langle \tilde{w}, z\|\bar{z}\rangle$ is a fair reach-trace of $y\|\bar{y}$. Let $w = \tilde{w}\downarrow A^V$ and $\bar{w} = \tilde{w}\downarrow \bar{A}^V$. Since the event $\tilde{W}^{(\tilde{J}_i)} = \tilde{w}$ implies $W^{(\tilde{J}_i)} = w \wedge \bar{W}^{(\tilde{J}_i)} = \bar{w}$, we have

$$\begin{aligned}0 &< \Pr(\tilde{X}_{\mathsf{post}}^{(\tilde{J}_i)} = z\|\bar{z}, \tilde{W}^{(\tilde{J}_i)} = \tilde{w} \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_i)} = y\|\bar{y})\\ &< \Pr(X_{\mathsf{post}}^{(\tilde{J}_i)} = z, \bar{X}_{\mathsf{post}}^{(\tilde{J}_i)} = \bar{z}, W^{(\tilde{J}_i)} = w, \bar{W}^{(\tilde{J}_i)} = \bar{w} \mid X_{\mathsf{pre}}^{(\tilde{J}_i)} = y, \bar{X}_{\mathsf{pre}}^{(\tilde{J}_i)} = \bar{y})\\ &< \Pr(X_{\mathsf{post}}^{(\tilde{J}_i)} = z, W^{(\tilde{J}_i)} = w \mid X_{\mathsf{pre}}^{(\tilde{J}_i)} = y, \bar{X}_{\mathsf{pre}}^{(\tilde{J}_i)} = \bar{y}).\end{aligned}$$

We then also have

$$\begin{aligned}0 &< \Pr(X^{(\tilde{J}_i)}_{\mathsf{post}} = z, W^{(\tilde{J}_i)} = w, X^{(\tilde{J}_i)}_{\mathsf{pre}} = y, \bar{X}^{(\tilde{J}_i)}_{\mathsf{pre}} = \bar{y})\\ &< \Pr(X^{(\tilde{J}_i)}_{\mathsf{post}} = z, W^{(\tilde{J}_i)} = w, X^{(\tilde{J}_i)}_{\mathsf{pre}} = y),\end{aligned}$$

which implies that

$$\Pr(X^{(\tilde{J}_i)}_{\mathsf{post}} = z, W^{(\tilde{J}_i)} = w \mid X^{(\tilde{J}_i)}_{\mathsf{pre}} = y) > 0.$$

Similarly we find

$$\Pr(\bar{X}^{(\tilde{J}_i)}_{\mathsf{post}} = \bar{z}, \bar{W}^{(\tilde{J}_i)} = \bar{w} \mid \bar{X}^{(\tilde{J}_i)}_{\mathsf{pre}} = \bar{y}) > 0.$$

Now, we know that $\tilde{J}_i$ must be either a jump time of X or of $\bar{X}$ or both. If $\tilde{J}_i$ is a jump-time of X then it follows from (6.4) that (w, z) is a fair reach-trace of y. If there is no jump for X at time $\tilde{J}_i$ then we have $W^{(J_i)} = \epsilon$ and $X^{(\tilde{J}_i)}_{\mathsf{pre}} = X^{(\tilde{J}_i)}_{\mathsf{post}}$. From Proposition 19 we have that y is a stable state, since

$$0 < \Pr(\tilde{X}^{(\tilde{J}_i)}_{\mathsf{pre}} = y \| \bar{y}) < \Pr(X^{(\tilde{J}_i)}_{\mathsf{pre}} = y) = \Pr(X^{(\tilde{J}_i)}_{\mathsf{post}} = y).$$

It then immediately follows that $(w, z) = (\epsilon, y)$ is a fair reach-trace of y. We then have that for both cases (w, z) is a fair reach-trace of y. Similarly we find that $(\bar{w}, \bar{z})$ is a fair-reach trace of $\bar{y}$ and then from Corollary 9 it follows that $(\tilde{w}, z \| \bar{z})$ is a fair-reach trace of $y \| \bar{y}$.

Requirement (6.5): Markovian jump. We will use the independence of Markovian jumps (and non-jumps) for X and $\bar{X}$ to show that (6.5) holds for $\tilde{X}$. Consider states $x, y \in S_\perp$, states $\bar{x}, \bar{y} \in \bar{S}_\perp$, a time-point $t \in \mathbb{R}_{\geq 0}$, and a time-length $h > 0$ such that $x \| \bar{x} \neq y \| \bar{y}$. There are then three possibilities for the projections of these states onto P and $\bar{P}$. Either, $x \neq y$, $\bar{x} \neq \bar{y}$, or $x \neq y, \bar{x} = \bar{y}$, or $x = y, \bar{x} \neq \bar{y}$.

For the first case ($x \neq y$ and $\bar{x} \neq \bar{y}$) we have

$$\begin{aligned}&\Pr(\tilde{J}^{(t)}_1 \leq t + h, \tilde{X}^{(\tilde{J}^{(t)}_1)}_{\mathsf{pre}} = y \| \bar{y} \mid \tilde{X}^{(t)}_{\mathsf{pre}} = x \| \bar{x})\\ &= \Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y, \bar{J}^{(t)}_1 \leq t + h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \mid X^{(t)}_{\mathsf{pre}} = x, \bar{X}^{(t)}_{\mathsf{pre}} = \bar{x}).\end{aligned}$$

Since X and $\bar{X}$ are compatible, their Markovian jump probabilities are "independent up to $o(h)$" due to (6.31). The above then equals

$$\begin{aligned}&\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = y \mid X^{(t)}_{\mathsf{pre}} = x)\\ &\qquad \cdot \Pr(\bar{J}^{(t)}_1 \leq t + h, \bar{X}^{(\bar{J}^{(t)}_1)}_{\mathsf{pre}} = \bar{y} \mid \bar{X}^{(t)}_{\mathsf{pre}} = \bar{x}) + o(h)\\ &= (q_{x,y} h + o(h))(\bar{q}_{\bar{x},\bar{y}} h + o(h)) + o(h) = o(h),\end{aligned}$$

which is what (6.5) prescribes as $\tilde{q}_{x\|\bar{x}, y\|\bar{y}} = 0$ whenever $x \neq y$ and $\bar{x} \neq \bar{y}$.

For the case $x \neq y$, $\bar{x} = \bar{y}$ we have a few possibilities for the time of the first jump of $\bar{X}$ after time t, i.e., $\bar{J}^{(t)}_1$. It may be that $\bar{J}^{(t)}_1$ is greater than $\tilde{J}^{(t)}_1$, but still occurs before

$t+h$. But this means that two jumps of $\tilde{X}$ occur within the time-interval $[t, t+h]$ and this occurs with probability $o(h)$ due to (6.28). Secondly, it may be that $\bar{J}_1^{(t)}$ is greater than $\tilde{J}_1^{(t)}$ and also greater than $t+h$. Finally, it may be that the jump at $\tilde{J}_1^{(t)}$ is also a jump of $\bar{X}$ (i.e., $\bar{J}_1^{(t)} = \tilde{J}_1^{(t)}$) and then we have $\bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x}$. Then, "up to $o(h)$" we have that either $\bar{J}_1^{(t)} > t+h$ or $\bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x}$. We then apply (6.32) to find

$$\begin{aligned}
&\Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = y\|\bar{y} \mid \tilde{X}_{\mathsf{pre}}^{(t)} = x\|\bar{x}) \\
&= \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y, (\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x}) \\
&\qquad\qquad \mid X_{\mathsf{pre}}^{(t)} = x, \bar{X}_{\mathsf{pre}}^{(t)} = \bar{x}) + o(h) \\
&= \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = y \mid X_{\mathsf{pre}}^{(t)} = x) \\
&\qquad \Pr(\bar{J}_1^{(t)} > t+h \vee \bar{X}_{\mathsf{pre}}^{(\bar{J}_1^{(t)})} = \bar{x} \mid \bar{X}_{\mathsf{pre}}^{(t)} = \bar{x}) + o(h) \\
&= (q_{x,y}h + o(h))(1 - \bar{q}_{\bar{x}}h + o(h)) + o(h) \\
&= q_{x,y}h + o(h),
\end{aligned}$$

which conforms to the fact that $\tilde{q}_{x\|\bar{x},y\|\bar{y}} = q_{x,y}$ whenever $x \neq y$ and $\bar{x} = \bar{y}$. The case $x = y$, $\bar{x} \neq \bar{y}$ proceeds in a symmetric fashion.

Requirement (6.6): Two jumps. This trivially follows from the fact that X and $\bar{X}$ are compatible and thus satisfy (6.28).

Requirement (6.7): local Markov property. This follows from the independence up to $o(h)$ of X and $\bar{X}$ and the fact that X and $\bar{X}$ themselves satisfy (6.7). □

A.1.11 Proof of Theorem 40

If $\tilde{X}$ has interactive jump scheduler $\tilde{\gamma}$, then we find for the interactive jump probabilities of behaviour X, that

$$\begin{aligned}
&\Pr(X_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y, W^{(\tilde{J}_{i+1})} \mid X_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = x, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma}) \\
&= \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \sum_{\bar{y} \in \bar{S}_\perp} \tilde{\gamma}_{\tilde{\sigma},x\|\bar{x}}^{(t)}(\tilde{w}, y\|\bar{y}),
\end{aligned} \tag{A.9}$$

for a jump-index $i \in \mathbb{N}_0$, a path $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, states $x, y \in S_\perp$, $\bar{x} \in \bar{S}_\perp$, a sequence $w \in \mathcal{L}^V$, and a time-point $t \in \mathbb{R}_{\geq 0}$. Moreover, for the function $f : \left(\{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}\right) \times \mathbb{R}_{\geq 0} \times \tilde{S}_\perp \times \mathcal{L}^V \times S_\perp \to [0,1]$ defined as

$$f(\tilde{\sigma}, t, x, w, y) \equiv \Pr(X_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y, W^{(\tilde{J}_{i+1})} \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = x\|\bar{x}, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma})$$

we find that

1. $f(\cdot,\cdot,\cdot,w,y)$ is a Borel-measurable function for fixed $w \in \mathcal{L}^V$ and $y \in S_\perp$,

2. $f(\tilde{\sigma}, t, \tilde{x}, \cdot, \cdot)$ is a probability function on $\mathcal{L}^V \times S_\perp$ for fixed $\tilde{\sigma} \in \{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, $t \in \mathbb{R}_{\geq 0}$, and $\tilde{x} \in \tilde{S}_\perp$, and

3. For any $\tilde{\sigma} \in \{\epsilon\} \cup \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, $t \in \mathbb{R}_{\geq 0}$, $x \in \tilde{S}_\perp$, $y \in S_\perp$, and $w \in \mathcal{L}^V$ we have $f(\sigma, t, \tilde{x}, w, y) > 0$ implies $(w, y) \in \mathit{FairRT}(\tilde{x}{\downarrow}P)$.

We find a similar result for the interactive jump probabilities of $\bar{X}$.

Proof. Given a path $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, states $\tilde{x} \in \tilde{S}_\perp$, a time-point $t \in \mathbb{R}_{\geq 0}$, a state $y \in S_\perp$, and a sequence $w \in \mathcal{L}^V$, we find

$$\begin{aligned}
&\Pr(X_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y, W^{(\tilde{J}_{i+1})} = w \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = \tilde{x}, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma}) \\
&= \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \sum_{\bar{y} \in \bar{S}_\perp} \Pr(\tilde{X}_{\mathsf{post}}^{(\tilde{J}_{i+1})} = y \| \bar{y}, \tilde{W}^{(\tilde{J}_{i+1})} = \tilde{w} \mid \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_{i+1})} = \tilde{x}, \tilde{J}_{i+1} = t, \tilde{Z}^{(\tilde{J}_i)} = \tilde{\sigma}).
\end{aligned}$$

Now (A.9) follows from the definition of the interactive jump scheduler of $\tilde{X}$. It remains to show that the function f induced by the interactive jump probabilities of X satisfies the three conditions from Theorem 40. For the first condition we find that the function $f(\cdot, \cdot, \cdot, w, y)$ is indeed Borel-measurable for fixed $w \in \mathcal{L}^V$ and $y \in S_\perp$, because $\tilde{\gamma}$ is Borel-measurable in the same way. To check whether $f(\tilde{\sigma}, t, \tilde{x}, \cdot, \cdot)$ is a probability function for fixed $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, $\tilde{x} \in \tilde{S}$, and $t \in \mathbb{R}_{\geq 0}$ we must simply check whether this function sums up to one. We have

$$\begin{aligned}
&\sum_{y \in S_\perp} \sum_{w \in \mathcal{L}^V} f(\tilde{\sigma}, t, \tilde{x}, w, y) \\
&= \sum_{y \in S_\perp} \sum_{w \in \mathcal{L}^V} \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \sum_{\bar{y} \in \bar{S}_\perp} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, y \| \bar{y}) \\
&= \sum_{y \| \bar{y} \in \tilde{S}_\perp} \sum_{\tilde{w} \in \tilde{\mathcal{L}}^V} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, y \| \bar{y}),
\end{aligned}$$

since for every sequence of actions $\tilde{w} \in \tilde{\mathcal{L}}^V$ we have that $\tilde{w}{\downarrow}P \in \mathcal{L}^V$. Now, from the fact that $\tilde{\gamma}^{(t)}_{\tilde{\sigma},x\|\bar{x}}$ is a probability function it follows that the above sum indeed equals one. Finally, we must make sure that f assigns positive probability only to fair reach-traces. That is, we must show that whenever

$$f(\tilde{\sigma}, t, \tilde{x}, w, y) > 0$$

we have $(w, y) \in \mathit{FairRT}(\tilde{x}{\downarrow}P)$. Now, considering the case $y \neq \perp$, the above implies that

$$\sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \sum_{\bar{y} \in \bar{S}_\perp} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, y \| \bar{y}) > 0$$

and then there exists some $\tilde{w} \in \tilde{\mathcal{L}}^V$ such that $\tilde{w}{\downarrow}P = w$ and some $\bar{y} \in \bar{S}_\perp$ such that

$$\tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, y \| \bar{y}) > 0.$$

Since $\tilde{\gamma}$ is an interactive jump scheduler for $\tilde{P}$ we have that $(\tilde{w}, y\|\bar{y})$ is a fair reach-trace of $IOA(\tilde{x})$. It then follows from the modularity of fair reach-traces (see Chapter 4) that the projection of $(\tilde{w}, y\|\bar{y})$ onto P, (w, y), is a fair reach-trace of $IOA(\tilde{x}\downarrow P)$. We find the same for the case $y = \perp$. □

A.1.12 Proof of Theorem 41

If $\tilde{X}$ has external jump scheduler $\tilde{\eta}$, then we find for the external jump probabilities of X that

$$\Pr(J_1^{(t)} \leq t + h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid \tilde{Z}^{(t)} = \tilde{\sigma}) = \left(\sum_{\substack{\bar{y} \in \bar{S}_\perp \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}^{(t)}_{\tilde{\sigma}} \right) h + o(h) \qquad \boxed{\text{A.10}}$$

for states $x \in S_\perp$, $\bar{x} \in \bar{S}_\perp$, a path $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$ with $\mathit{last}(\tilde{\sigma}) = x\|\bar{x}$ and a time-point $t \in \mathbb{R}_{\geq 0}$. Moreover, we have that the function $f : \mathit{FinPaths}_{\tilde{S},\tilde{A}} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$f(\tilde{\sigma}, t) = \begin{cases} \sum_{\substack{\bar{y} \in \bar{S}_\perp \\ \bar{y} \neq \bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}^{(t)}_{\tilde{\sigma}}, & \text{if } \mathit{last}(\tilde{\sigma}) = x\|\bar{x} \\ 0, & \text{if } \mathit{last}(\tilde{\sigma}) = \perp. \end{cases}$$

is Borel-measurable. We find a similar result for the external jump scheduler $\bar{\eta}$ of $\bar{X}$.

Proof. For a timed-path $\tilde{\sigma} \in \mathit{FinPaths}_{\tilde{S},\tilde{A}}$, a state $x \in S_\perp$, and time-points $t < t + h \in \mathbb{R}_{\geq 0}$ we have for the external jump probability of X that

$$\begin{aligned} &\Pr(J_1^{(t)} \leq t + h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid \tilde{Z}^{(t)} = \tilde{\sigma}) \\ &= \sum_{\bar{y} \in \bar{S}_\perp} \Pr(\tilde{J}_1^{(t)} \leq t + h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = x\|\bar{y} \mid \tilde{Z}^{(t)} = \tilde{\sigma}) + o(h) \end{aligned}$$

since a jump of X implies a jump of $\tilde{X}$ and the probability that $J_1^{(t)}$ is not the first jump of $\tilde{X}$ after t is $o(h)$ because of (6.6). The jump at time $\tilde{J}_1^{(t)}$ may be either Markovian or external depending on whether $\bar{y}$ is equal to $\bar{x}$ or not. We then have that the above

equals

$$\begin{aligned}
&\sum_{\substack{\bar{y}\in\bar{S}_\perp\\ \bar{y}\neq\bar{x}}} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = x\|\bar{y} \mid \tilde{Z}^{(t)} = \tilde{\sigma}) \\
&+ \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = x\|\bar{x} \mid \tilde{Z}^{(t)} = \tilde{\sigma}) + o(h) \\
&= \sum_{\substack{\bar{y}\in\bar{S}_\perp\\ \bar{y}\neq\bar{x}}} \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = x\|\bar{y} \mid \tilde{X}_{\mathsf{post}}^{(t)} = x\|\bar{x}) \\
&+ \Pr(\tilde{J}_1^{(t)} \leq t+h, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_1^{(t)})} = x\|\bar{y} \mid \tilde{Z}^{(t)} = \tilde{\sigma}) + o(h),
\end{aligned}$$

where we applied (6.7). Now we can apply (6.5) and the definition of the external jump scheduler to find that the above equals

$$\sum_{\substack{\bar{y}\in\bar{S}_\perp\\ \bar{y}\neq\bar{x}}} \tilde{q}_{x\|\bar{x},x\|\bar{y}} h + \tilde{\eta}_{\tilde{\sigma}}^{(t)} h + o(h) = \left(\sum_{\substack{\bar{y}\in\bar{S}_\perp\\ \bar{y}\neq\bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}_{\tilde{\sigma}}^{(t)} \right) h + o(h),$$

where we applied (5.3). It remains to show that the function f is Borel-measurable, but this follows directly from the Borel-measurability of $\tilde{\eta}$. □

A.1.13 Proof of Proposition 18

Let X, $\bar{X}$, and $\tilde{X}$ be behaviours of P, $\bar{P}$, and $\tilde{P} = P\|\bar{P}$ respectively and let $\tilde{\gamma}$ and $\tilde{\eta}$ be the interactive jump respectively external jump scheduler of $\tilde{X}$. Given a path $\sigma \in \mathit{FinPaths}_{S,A}$, a time-point $t \in \mathbb{R}_{\geq 0}$, states $x, y \in S_\perp$, a sequence of actions $w \in \mathcal{L}^V$, and a jump-index $i \in \mathbb{N}_0$ we find for the interactive jump probabilities of X that

$$\begin{aligned}
&\Pr(X_{\mathsf{post}}^{(J_{i+1})} = y, W^{(J_{i+1})} = w \mid X_{\mathsf{pre}}^{(J_{i+1})} = x, J_{i+1} = t, Z^{(J_i)} = \sigma) \\
&= \sum_{k=i}^{\infty} \sum_{\substack{\tilde{x}\in\tilde{S}_\perp\\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma}\in H_k} \sum_{\substack{\tilde{y}\in\tilde{S}_\perp\\ \tilde{y}\downarrow P = y}} \sum_{\substack{\tilde{w}\in\tilde{\mathcal{L}}^V\\ \tilde{w}\downarrow P = w}} \tilde{\gamma}_{\tilde{\sigma},\tilde{x}}^{(t)}(\tilde{w},\tilde{y}) \\
&\quad \cdot \Pr(\tilde{J}_{k+1} = J_{i+1}, \tilde{X}_{\mathsf{pre}}^{(\tilde{J}_{k+1})} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \mid X_{\mathsf{pre}}^{(J_{i+1})} = x, J_{i+1} = t, Z^{(J_i)} = \sigma). \qquad \text{(A.11)}
\end{aligned}$$

Furthermore, given a path $\sigma \in \mathit{FinPaths}_{S,A}$, a time-point $t \in \mathbb{R}_{\geq 0}$, and a state $x \in S_\perp$, we find for the external jump probabilities of X that

$$\begin{aligned}
&\Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x \mid Z^{(t)} = \sigma) \\
&= \left(\sum_{\bar{x}\in\bar{S}} \int_{\substack{\tilde{\sigma}\in H\\ \mathit{last}(\tilde{\sigma})=x\|\bar{x}}} \left(\sum_{\substack{\bar{y}\in\bar{S}\\ \bar{y}\neq\bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}_{\tilde{\sigma}}^{(t)} \right) \Pr(\tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma) \right) h + o(h). \qquad \text{(A.12)}
\end{aligned}$$

Proof. We start by expressing the jump probability of P from Equation (A.11) in terms of the jumps of $\tilde{P}$. This leads to

$$\begin{aligned}
&\Pr(X^{(J_{i+1})}_{\mathsf{post}} = y, W^{(J_{i+1})} = w \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma) \\
&= \sum_{k=i}^{\infty} \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma} \in H_k} \\
&\Pr(X^{(J_{i+1})}_{\mathsf{post}} = y, W^{(J_{i+1})} = w, \tilde{J}_{k+1} = J_{i+1}, \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \\
&\qquad \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma),
\end{aligned}$$

where $H_k \subset \mathit{Paths}^{(k)}_{\tilde{S},\tilde{A}}$ is the set of all timed paths $\tilde{P}$ of length k whose projection onto P is in the set of paths $d\sigma$. Now that we have fixed the history of $\tilde{X}$ we can express the interactive jump probability of X in terms of the "scheduler" from Theorem 40. We have that the above equals

$$\begin{aligned}
&\sum_{k=i}^{\infty} \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma} \in H_k} \Pr(X^{(J_{i+1})}_{\mathsf{post}} = y, W^{(J_{i+1})} = w \mid \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{J}_{k+1} = t, \tilde{Z}^{(\tilde{J}_k)} = \tilde{\sigma}) \\
&\quad \cdot \Pr(\tilde{J}_{k+1} = J_{i+1}, \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma) \\
&= \sum_{k=i}^{\infty} \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\tilde{\sigma} \in H_k} \sum_{\substack{\tilde{y} \in \tilde{S}_\perp \\ \tilde{y}\downarrow P = y}} \sum_{\substack{\tilde{w} \in \tilde{\mathcal{L}}^V \\ \tilde{w}\downarrow P = w}} \tilde{\gamma}^{(t)}_{\tilde{\sigma},\tilde{x}}(\tilde{w}, \tilde{y}) \\
&\quad \cdot \Pr(\tilde{J}_{k+1} = J_{i+1}, \tilde{X}^{(\tilde{J}_{k+1})}_{\mathsf{pre}} = \tilde{x}, \tilde{Z}^{(\tilde{J}_k)} \in d\tilde{\sigma} \mid X^{(J_{i+1})}_{\mathsf{pre}} = x, J_{i+1} = t, Z^{(J_i)} = \sigma).
\end{aligned}$$

Similarly, we find for the external jump probabilities of X that, given a path $\sigma \in \mathit{FinPaths}_{S,A}$, a time-point $t \in \mathbb{R}_{\geq 0}$, a time length $h > 0$, and a state $x \in S_\perp$, we have

$$\begin{aligned}
&\Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = x \mid Z^{(t)} = \sigma) \\
&= \sum_{\substack{\tilde{x} \in \tilde{S}_\perp \\ \tilde{x}\downarrow P = x}} \int_{\substack{\tilde{\sigma} \in H \\ \mathit{last}(\tilde{\sigma}) = \tilde{x}}} \Pr(J^{(t)}_1 \leq t + h, X^{(J^{(t)}_1)}_{\mathsf{pre}} = x, \tilde{X}^{(t)}_{\mathsf{post}} = \tilde{x}, \tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma),
\end{aligned}$$

where $H \subset \mathit{FinPaths}_{\tilde{S},\tilde{A}}$ is the set of all timed paths of $\tilde{P}$ whose projection onto P is in $d\sigma$. For the case $\tilde{x} = \perp$ we have that the probability of any further jump is zero by definition. We then have that $\tilde{x}$ must be the parallel composition of x with some state

$\bar{x}$ and we find that the above equals

$$\sum_{\bar{x}\in\bar{S}} \int_{\substack{\tilde{\sigma}\in H \\ last(\tilde{\sigma})=x \| \bar{x}}} \Pr(J_1^{(t)} \leq t+h, X_{\mathsf{pre}}^{(J_1^{(t)})} = x, \mid \tilde{Z}^{(t)} = \tilde{\sigma}) \Pr(\tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma)$$
$$= \left(\sum_{\bar{x}\in\bar{S}} \int_{\substack{\tilde{\sigma}\in H \\ last(\tilde{\sigma})=x \| \bar{x}}} \left(\sum_{\substack{\bar{y}\in\bar{S} \\ \bar{y}\neq\bar{x}}} \bar{q}_{\bar{x},\bar{y}} + \tilde{\eta}_{\tilde{\sigma}}^{(t)} \right) \Pr(\tilde{Z}^{(t)} \in d\tilde{\sigma} \mid Z^{(t)} = \sigma) \right) h + o(h).$$

□

A.1.14 Proof of Proposition 23

Given a behaviour X of P, let $\bar{X} = X \backslash B$. For any time-point $t \in \mathbb{R}_{\geq 0}$, we then have

$$\Pr(\bar{J}_1^{(t)} \neq J_1^{(t)}) = 0.$$

Proof. First, we note that $\bar{X}_{\mathsf{pre}} = X_{\mathsf{pre}}$, $\bar{X}_{\mathsf{post}} = X_{\mathsf{post}}$, and $\bar{W} = W \downarrow (A^V \setminus B)$. It follows that whenever there is a jump of $\bar{X}$, there is also a jump for X. And whenever there is a jump for X there is a jump for $\bar{X}$, except if $X_{\mathsf{pre}}^{(J_1^{(t)})} = X_{\mathsf{post}}^{(t)}$, $X_{\mathsf{post}}^{(J_1^{(t)})} = X_{\mathsf{pre}}^{(J_1^{(t)})}$, and the sequence of actions for this jump contains only actions from B, since then $W \downarrow (A^V \setminus B)$ will equal ϵ while W itself does not. Assume then that

$$\Pr(\bar{J}_1^{(t)} \neq J_1^{(t)}) > 0.$$

It follows that there exists a state x and a sequence of actions $w \in B^*$ such that

$$\Pr(\bar{J}_1^{(t)} \neq J_1^{(t)}, X^{(J_1^{(t)})} = (x, w, x), X_{\mathsf{post}}^{(t)} = x) > 0.$$

Because of requirement (6.4) we have that (w, x) must be a fair reach-trace of x. Since w contains no input actions, this means that the state x is unstable. But then we find, due to Proposition 19 that

$$\Pr(X_{\mathsf{post}}^{(t)} = x) = 0.$$

This is a contradiction. □

A.1.15 Proof of Theorem 42

Given an I/O-IMC P, a subset of its output actions B, and a behaviour X of P, we have that $X \backslash B$ is a behaviour of $P \backslash B$.

Proof. Let $\bar{X}$ be the interactive jump process $X \backslash B$. We have $\bar{X}_{\mathsf{pre}} = X_{\mathsf{pre}}$ and $\bar{X}_{\mathsf{post}} = X_{\mathsf{post}}$ and thus it trivially follows that requirements (6.3), (6.5), (6.6), (6.7) hold for $\bar{X}$ as they do not involve the stochastic process $\bar{W}$ which differs from W. It then remains

to show that $\bar{X}$ satisfies (6.4). Consider a jump-index $i \in \mathbb{N}_0$, states $y, z \in S_\perp$, and a sequence of actions $\bar{w} \in \bar{\mathcal{L}}^V$ such that $\Pr(\bar{X}^{(\bar{J}_i)}_{\mathsf{pre}} = y) > 0$. Now assume that

$$\Pr(\bar{X}^{(\bar{J}_i)}_{\mathsf{post}} = z, \bar{W}^{(\bar{J}_i)} = \bar{w} \mid \bar{X}^{(\bar{J}_i)}_{\mathsf{pre}} = y) > 0.$$

Since the jumps of $\bar{X}$ correspond to the jumps of X with probability one we have

$$\sum_{\substack{w \in \mathcal{L}^V: \\ w \backslash B = \bar{w}}} \Pr(X^{(J_i)}_{\mathsf{post}} = z, W^{(J_i)} = w \mid X^{(J_i)}_{\mathsf{pre}} = y) > 0$$

and then there exists a sequence $w \in \mathcal{L}^V$ with $w \backslash B = \bar{w}$ such that

$$\Pr(X^{(J_i)}_{\mathsf{post}} = z, W^{(J_i)} = w \mid X^{(J_i)}_{\mathsf{pre}} = y) > 0.$$

It follows that (w, z) is a fair reach-trace of y in the I/O-IMC P. From Proposition 7 it then follows that $(\bar{w}, z)$ if a fair reach-trace of y in the I/O-IMC $P \backslash B$. □

A.1.16 Proof of Theorem 43

Given an I/O-IMC P, a subset of its output actions B, and a behaviour $\bar{X}$ of $P \backslash B$, we have that there exists a behaviour X of P such that $\bar{X} = X \backslash B$.

Proof. Given that the behaviour $\bar{X}$ has probability space $(Paths_{S,\bar{A}}, \mathcal{F}_{S,\bar{A}}, \bar{\mathcal{P}})$, we must construct a probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ such that the associated interactive jump process is a behaviour of P. We choose the sets of timed-paths $Paths_{S,\bar{A}}$ respectively $Paths_{S,A}$ and the σ-algebras $\mathcal{F}_{S,\bar{A}}$ respectively $\mathcal{F}_{S,\bar{A}}$ as in Subsection 6.2. It then remains to construct the probability function $\mathcal{P}$. We will do this by providing a function f from $\mathcal{F}_{S,A}$ to $\mathcal{F}_{S,\bar{A}}$ which will allow us to derive the probability function $\mathcal{P}$ from $\bar{\mathcal{P}}$.

First, we associate the interactive jumps of $\bar{X}$ with interactive jumps in P. Let $g : S_\perp \times \bar{\mathcal{L}}^V \times S_\perp \to \mathcal{L}^V$ be any partial function which satisfies the following requirements. For all states $x, y \in S_\perp$ and sequence $\bar{w} \in \bar{\mathcal{L}}^V$ we have that if $(\bar{w}, y)$ is a fair reach-trace of x in $P \backslash B$ then for the sequence $w = g(x, \bar{w}, y)$ we have $\bar{w} = w \backslash B$ and (w, y) is a fair-reach trace of x in P. On the other hand if $(\bar{w}, y)$ is not a fair reach-trace of x in $P \backslash B$, then $g(x, \bar{w}, y)$ is undefined. Proposition 7 ensures that we can always find such a function g, since for every fair reach-trace in $P \backslash B$ there exists a corresponding fair-reach trace for P.

Now, we will define our function f from $\mathcal{F}_{S,A}$ to $\mathcal{F}_{S,\bar{A}}$ as follows. Fixing two equal-length, possibly infinite, sequences of states in $S_\perp$, $x_0, x_1, \ldots$ and $y_0, y_1, \ldots$ and a sequence of action-sequences $w_0, w_1, \ldots$ in $\mathcal{L}^V$ also of the same length, let H equal the set of timed-paths

$$\{(x_0, w_0, y_0, t_1, x_1, w_1, y_1, \ldots) \mid t_1 \in [s_1, s_1'], t_2 \in [s_2, s_2']\},$$

for some sequence of time-bounds $s_1, s_1', s_2, s_2' \in \mathbb{R}_{\geq 0}$. We then define

$$f(H) = H',$$

where

$$H' = \begin{cases} \{(x_0, w_0 \backslash B, y_0, t_1, x_1, w_1 \backslash B, y_1, \ldots) \mid t_1 \in [s_1, s_1'], t_2 \in [s_2, s_2']\}, \\ \qquad \text{if } g(x_0, w_0 \backslash B, y_0) = w_0, g(x_1, w_1 \backslash B, y_1) = w_1, \ldots \\ \emptyset, \text{ otherwise.} \end{cases}$$

Furthermore we have that f is commutative with respect to countable union. That is, for measurable sets $H_1, H_2, \ldots \in \mathcal{F}_{S,A}$ we have

$$f(H_1 \cup H_2 \cup \ldots) = f(H_1) \cup f(H_2) \cup \ldots.$$

It can be verified that this completely defines the function f.

Now we are ready to define $\mathcal{P}$. We have

$$\mathcal{P}(H) = \bar{\mathcal{P}}(f(H)).$$

Since f is commutative with respect to countable union and since $\bar{\mathcal{P}}$ is a probability function we find that $\mathcal{P}$ is also a probability function.

It is clear that the stochastic process X with probability space $(Paths_{S,A}, \mathcal{F}_{S,A}, \mathcal{P})$ as defined above is a stable interactive jump process of P. It then remains to show that it is also a behaviour of P. Since we have $X_{\mathsf{pre}} = \bar{X}_{\mathsf{pre}}$ and $X_{\mathsf{post}} = \bar{X}_{\mathsf{post}}$ and the jumps of X and $\bar{X}$ coincide with probability one, it follows that requirements (6.3), (6.5), (6.6), and (6.7) hold for X. It remains to show that (6.4) holds.

$$\Pr(X_{\mathsf{post}}^{(J_i)} = y, W^{(J_i)} = w \mid X_{\mathsf{pre}}^{(J_i)} = x) > 0$$

implies $(w, y) \in \mathit{FairRT}(x)$. □

A.2 Proofs of Chapter 7

A.2.1 Proof of Proposition 25

Given a behaviour X of closed I/O-IMC P we have that if X is non-divergent then X is closed.

Proof. We prove Proposition 25 by contradiction. We then assume that X is non-divergent, but not closed. This means there exists some jump-index $i \in \mathbb{N}_0$ and state $x \in S_\perp$ such that $\Pr(X_{\mathsf{post}}^{(J_i)} = x) > 0$ and $\Pr(X_{\mathsf{pre}}^{(J_{i+1})} = x \mid X_{\mathsf{post}}^{(J_i)} = x) > 0$. Now, due to Proposition 19 we know that the state x must be stable. This means that x has no outgoing output or internal transitions. Since P is closed and has no input action, x obviously also has no outgoing input transitions. We then have that x has no outgoing interactive transitions at all and $\mathit{FairRT}(x) = \{(\epsilon, x), (\epsilon, \perp)\}$. Moreover, since x has no outgoing interactive transitions we have that x cannot interactively reach any divergent states.

Let's consider the jump at time J_{i+1}. We have $\Pr(X_{\mathsf{pre}}^{(J_{i+1})} = x) > 0$ and then we must find some $w \in \mathcal{L}^V$ and $y \in S_\perp$ such that

$$\Pr(X_{\mathsf{post}}^{(J_{i+1})} = y, W^{(J_{i+1})} = w \mid X_{\mathsf{pre}}^{(J_{i+1})} = x) > 0.$$

Since, X is non-divergent and we cannot reach a divergent state from x we have $y \neq \perp$. However, because of (6.4) (w, y) must be a fair-reach trace of x. The only possibility is then that $w = \epsilon$ and $y = x$, but this means that, with probability greater than zero, no jump occurred at J_{i+1}, which is a contradiction with the definition of jump-times. □

A.2.2 Proof of Theorem 45

Given a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ and a weak bisimulation relation $\mathcal{E}$ on S, we find for any equivalence class $D \in S/\mathcal{E}$, any time-point $t \in \mathbb{R}_{\geq 0}$, and any jump-index $i \in \mathbb{N}$, that

$$\Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D) = 1 - e^{-\bar{q}_D t}.$$

Proof. As usual, we will find this cumulative probability by first determining its derivative

$$\frac{d}{dt} \Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D) = \lim_{h \downarrow 0} \Pr(t < K_{i+1} \leq t + h \mid X^{(K_i)}_{\mathsf{post}} \in D)/h.$$

We find

$$\begin{aligned} &\Pr(t < K_{i+1} \leq t + h \mid X^{(K_i)}_{\mathsf{post}} \in D) \\ &\quad = \sum_{x \in D} \Pr(t < K_{i+1} \leq t + h, X^{(t)}_{\mathsf{post}} = x \mid X^{(K_i)}_{\mathsf{post}} \in D), \end{aligned}$$

since the absence of an ϵ-jump between K_i and t must mean that X_{post} still occupies a state in D at time t. It's clear that there must occur at least one jump between equivalence classes between time-points t and $t + h$. The first of these jumps occurs at time K_{i+1} (by definition) and will start in state x. Recall from our definition of the equivalence-jump-times K, that for this jump we have either $X^{(K_{i+1})}_{\mathsf{pre}} \notin D$ (a Markovian jump to a state outside D), or $W^{(K_{i+1})} \neq \epsilon$ (an interactive jump which includes visible actions), or $X^{(K_{i+1})}_{\mathsf{post}} \notin D$ (an interactive jump that ends up in a different equivalence. Now we know that x is stable since the equivalence class D is stable. This means that an interactive jump (cases 2 and 3) can only occur if we first see a Markovian jump to an unstable state y. But this state y cannot lie in equivalence class D (which is stable). In the end this means that we will always have $X^{(K_{i+1})}_{\mathsf{pre}} \notin D$, for the $i+1$-th jump. We

can then rewrite the above as

$$\begin{aligned}
&\sum_{x\in D} \Pr(K_{i+1} \leq t+h \mid X^{(t)}_{\mathsf{post}} = x) \\
&\qquad \Pr(X^{(K_i)}_{\mathsf{post}} = x \mid K_{i+1} > t, X^{(K_i)}_{\mathsf{post}} \in D) \Pr(K_{i+1} > t \mid X^{(K_i)}_{\mathsf{post}} \in D) \\
=&\sum_{x\in D} \sum_{y\notin D} \Pr(X^{(K_{i+1})}_{\mathsf{pre}} = y, K_{i+1} \leq t+h \mid X^{(t)}_{\mathsf{post}} = x) \\
&\qquad \Pr(X^{(K_i)}_{\mathsf{post}} = x \mid K_{i+1} > t, X^{(K_i)}_{\mathsf{post}} \in D) \Pr(K_{i+1} > t \mid X^{(K_i)}_{\mathsf{post}} \in D) \\
=&\sum_{x\in D} \sum_{y\notin D} (q_{xy}h + o(h)) \\
&\qquad \Pr(X^{(K_i)}_{\mathsf{post}} = x \mid K_{i+1} > t, X^{(K_i)}_{\mathsf{post}} \in D) \Pr(K_{i+1} > t \mid X^{(K_i)}_{\mathsf{post}} \in D)
\end{aligned}$$

since a jump between equivalence classes implies a jump between states and the probability of more than one jump between states is $o(h)$. Now we consider the fact that all states in D have the same outgoing Markovian transitions (up to $\mathcal{E}$) and find:

$$\begin{aligned}
&\sum_{D'\neq D} (\bar{q}_{D,D'}h + o(h)) \\
&\qquad \sum_{x\in D} \Pr(X^{(K_i)}_{\mathsf{post}} = x \mid K_{i+1} > t, X^{(K_i)}_{\mathsf{post}} \in D) \Pr(K_{i+1} > t \mid X^{(K_i)}_{\mathsf{post}} \in D) \\
=&(\bar{q}_D h + o(h)) \Pr(K_{i+1} > t \mid X^{(K_i)}_{\mathsf{post}} \in D) \\
=&(\bar{q}_D h + o(h))(1 - \Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D)).
\end{aligned}$$

For the derivative of the distribution of K_{i+1} we then have

$$\frac{d}{dt} \Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D) = \bar{q}_D - \bar{q}_D \Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D).$$

We can solve this differential equation to find

$$\Pr(K_{i+1} \leq t \mid X^{(K_i)}_{\mathsf{post}} \in D) = 1 - e^{-\bar{q}_D t},$$

where we use the fact that this probability must be zero when t equals zero.

Note that our proof is considerably simpler than the same proof for CTMCs presented in Section 3.2, because we do not consider the possibility of infinitely many jumps occurring in finite time. This is justified by the fact that we only consider finite I/O-IMCs. □

A.2.3 Proof of Theorem 47

For every interactive jump scheduler γ of P there exists an interactive jump scheduler $\bar{\gamma}$ for $\bar{P}$ such that, for the induced behaviours X respectively $\bar{X}$ we have, for any time-point t, that

$$\Pr(X^{(t)}_{\mathsf{post}} \in D \cap S) = \Pr(\bar{X}^{(t)}_{\mathsf{post}} \in D \cap \bar{S}).$$

Proof. We will consider the history processes Z and $\bar{Z}$ of X and $\bar{X}$ respectively. We will prove Theorem 47 by proving that, for an index $n \in \mathbb{N}$, a set of equivalence class paths of length n of the form

$$H_n = D_1 \times \{w_1\} \times D_1' \times (0,\infty) \times D_2 \times \{w_2\} \times D_2' \times (0,\infty), \dots \tag{A.13}$$

(i.e., a set of paths that jumps from one equivalence class to another and which does not restrict on the jump times), equivalence classes D and D' of $\mathcal{E}$, and a word $w \in \mathcal{L}^V$, we have

$$\Pr(Z^{(t)} \in H_n \times (0,\infty) \times D \times \{w\} \times D') = \Pr(\bar{Z}^{(t)} \in H_n \times (0,\infty) \times D \times \{w\} \times D'). \tag{A.14}$$

We will prove this by induction over the number of jumps n.

We will need one more piece of notation for this proof. For a given finite path σ (of either P or $\bar{P}$) we will denote the *expansion* of σ with respect to the equivalence relation $\mathcal{E}$ as $[\sigma]_{\mathcal{E}}$, i.e. for a path

$$\sigma = x_1, w_1, y_1, t_1, x_2, w_2, y_2, t_2, \dots$$

we have

$$[\sigma]_{\mathcal{E}} = D_1 \times \{w_1\} \times D_1' \times (0,\infty) \times D_2 \times \{w_2\} \times D_2' \times (0,\infty), \dots,$$

where $x_1 \in D_1, y_1 \in D_1'$ and so forth.

Before we begin with the induction, we must first choose how to define the interactive jump scheduler $\bar{\gamma}$ for $\bar{P}$. To make matters a bit easier, we order all states in $\bar{S}$ in an arbitrary way and we will often refer to the *first* state in a subset of $\bar{S}$ according to this order. Now, given a path $\bar{\sigma}$, with $n \in \mathbb{N}$ jumps between equivalence classes, that ends in equivalence class D_1, a state $\bar{y} \in D_2 \cap \bar{S}$, a word $w \in \mathcal{L}^V$, and the *first* state $\bar{z} \in D_3 \cap \bar{S}$ such that there is an interactive path from $\bar{y}$ to $\bar{z}$ with word w, we define the interactive scheduler $\bar{\gamma}$ as follows

$$\bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}(w,\bar{z}) = \sum_{x \in D_1 \cap S} \sum_{y \in D_2 \cap S} \sum_{z \in D_3 \cap S} \int_{\substack{\sigma \in [\bar{\sigma}]_{\mathcal{E}} \\ last(\sigma) = x}} \Pr(Z^{(t)} \in d\sigma \,|\, Z^{(t)} \in [\bar{\sigma}]_{\mathcal{E}}) \frac{q_{x,y}}{q_{x,D_2}} \gamma^{(t)}_{\sigma,y}(w,z). \tag{A.15}$$

The above equation represents the probability that P makes an interactive jump with word w to a state in D_3 under the condition that we first follow a path which is equivalent to $\bar{\sigma}$ up to the equivalence classes of $\mathcal{E}$. For all other states $\bar{z}'$ in $D_3 \cap \bar{S}$ we choose

$$\bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}(w,\bar{z}') = 0.$$

For the special case of $t = 0$ we choose

$$\bar{\gamma}^{(0)}_{\epsilon,\bar{y}}(w,\bar{z}) = \sum_{y \in D \cap S} \sum_{z \in D_3 \cap S} \frac{\alpha(y)}{\alpha(D \cap S)} \gamma^{(0)}_{\epsilon,y}(w,z)$$

for the first $\bar{z}$ in $D_3 \cap S$ such that there is an interactive path from $\bar{y}$ to $\bar{z}$ with word w and zero for other states in $D_3 \cap S$.

It's important that we show that $\bar{\gamma}$ is in fact an interactive jump scheduler of P, i.e., it must satisfy the conditions in Definition 75. For fixed w and $\bar{z}$ we must show that $\bar{\gamma}^{(\cdot)}_{\cdot,\cdot}$ is a Borel-measurable function. We will leave the question whether this is the case for our definition of $\bar{\gamma}$ open, but it is clear that we can choose (in the usual way) a $\bar{\gamma}$ which *is* Borel-measurable and arbitrarily close to A.15. The second condition of Definition 75 states that $\bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}$ must be a probability function (i.e., must sum up to one). For fixed t, $\bar{\sigma}$ ending in $D_1 \cap \bar{S}$, and $\bar{y} \in D_2 \cap \bar{S}$, we have

$$\begin{aligned}&\sum_{w \in \mathcal{L}^V} \sum_{\bar{z} \in \bar{S}} \bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}(w, \bar{z}) \\ &= \sum_{w \in \mathcal{L}^V} \sum_{D_3 \in S \cup \bar{S}/\mathcal{E}} \sum_{\bar{z} \in D_3 \cap \bar{S}} \bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}(w, \bar{z}).\end{aligned}$$

For the equivalence class D_3 we must distinguish two cases, either there is an interactive path from $\bar{y}$ to a state in D_3 with word w or not. For the former case we will find a state $\bar{z} \in D_3 \cap \bar{S}$ such that it is the *first* state with such a path, for the latter case we of course won't find such a state and then $\sum_{\bar{z} \in D_3 \cap \bar{S}} \bar{\gamma}^{(t)}_{\bar{\sigma},\bar{y}}(w, \bar{z}) = 0$. Let $mathcal D_{\bar{y},w}$ denote the set of equivalence classes such that there is an interactive path from $\bar{y}$ to the equivalence class with word w. We now have that the above equals

$$\begin{aligned}&\sum_{w \in \mathcal{L}^V} \sum_{D_3 \in \mathcal{D}_{\bar{y},w}} \sum_{x \in D_1 \cap S} \sum_{y \in D_2 \cap S} \sum_{z \in D_3 \cap S} \int_{\substack{\sigma \in [\bar{\sigma}]_\mathcal{E} \\ last(\sigma) = x}} \Pr(Z^{(t)} \in d\sigma \,|\, Z^{(t)} \in [\bar{\sigma}]_\mathcal{E}) \frac{q_{x,y}}{q_{x,D_2}} \gamma^{(t)}_{\sigma,y}(w, z) \\ &= \sum_{x \in D_1 \cap S} \sum_{y \in D_2 \cap S} \int_{\substack{\sigma \in [\bar{\sigma}]_\mathcal{E} \\ last(\sigma) = x}} \Pr(Z^{(t)} \in d\sigma \,|\, Z^{(t)} \in [\bar{\sigma}]_\mathcal{E}) \frac{q_{x,y}}{q_{x,D_2}} \sum_{w \in \mathcal{L}^V} \sum_{D_3 \in \mathcal{D}_{\bar{y},w}} \sum_{z \in D_3 \cap S} \gamma^{(t)}_{\sigma,y}(w, z),\end{aligned}$$

For a state $z \in D' \notin \mathcal{D}_{\bar{y},w}$ we have that there is no interactive path from $\bar{y}$ to D' with word w. Now, since y and $\bar{y}$ are equivalent according to $\mathcal{E}$ (they are both in D_2) we have that y also does not have an interactive path to equivalence class D' with word w. It follows that $\gamma^{(t)}_{\sigma,y}(w, z') = 0$ and $\sum_{w \in \mathcal{L}^V} \sum_{D_3 \in \mathcal{D}_{\bar{y},w}} \sum_{z \in D_3 \cap S} \gamma^{(t)}_{\sigma,y}(w, z) = \sum_{w \in \mathcal{L}^V} \sum_{z \in S} \gamma^{(t)}_{\sigma,y}(w, z)$. We then find that the above equals

$$\sum_{x \in D_1 \cap S} \sum_{y \in D_2 \cap S} \int_{\substack{\sigma \in [\bar{\sigma}]_\mathcal{E} \\ last(\sigma) = x}} \Pr(Z^{(t)} \in d\sigma \,|\, Z^{(t)} \in [\bar{\sigma}]_\mathcal{E}) \frac{q_{x,y}}{q_{x,D_2}},$$

since $\gamma^{(t)}_{\sigma,y}$ must be a probability function itself. We can further simplify to

$$\sum_{x \in D_1 \cap S} \int_{\substack{\sigma \in [\bar{\sigma}]_\mathcal{E} \\ last(\sigma) = x}} \Pr(Z^{(t)} \in d\sigma \,|\, Z^{(t)} \in [\bar{\sigma}]_\mathcal{E}) = 1.$$

For the special case $t = 0$ we have, given a fixed state $\bar{y} \in D$,

$$\sum_{w \in \mathcal{L}^V} \sum_{\bar{z} \in \bar{S}} \bar{\gamma}^{(0)}_{\epsilon,\bar{y}}(w, \bar{z}) = \sum_{w \in \mathcal{L}^V} \sum_{D_3 \in S \cup \bar{S}/\mathcal{E}} \sum_{y \in D \cap S} \sum_{z \in D_3 \cap S} \frac{\alpha(y)}{\alpha(D \cap S)} \gamma^{(0)}_{\epsilon,y}(w, z)$$
$$= \sum_{y \in D \cap S} \frac{\alpha(y)}{\alpha(D \cap S)} = 1.$$

The final condition of Definition 75 states that the scheduler probabilities $\bar{\gamma}^{t)}_{\sigma,\bar{y}}(w, \bar{z})$ may be non-zero only if there is a path from $\bar{y}$ to $\bar{z}$ with word w, but this follows directly from our definition of $\bar{\gamma}$.

We can now proceed to show by induction on n that A.14 holds. We first consider the case $n = 0$, i.e.,

$$\Pr(Z^{(t)} \in D \cap S \times \{w\} \times D' \cap S) = \Pr(\bar{Z}^{(t)} \in D \cap \bar{S} \times \{w\} \times D' \cap \bar{S}).$$

For $t = 0$ we have

$$\Pr(\bar{Z}^{(t)} \in D \cap \bar{S} \times \{w\} \times D' \cap \bar{S}) = \sum_{\bar{y} \in D \cap \bar{S}} \sum_{\bar{z} \in D' \cap \bar{S}} \bar{\alpha}(\bar{y}) \bar{\gamma}^{(0)}_{\epsilon,\bar{y}}(w, \bar{z})$$
$$= \sum_{\bar{y} \in D \cap \bar{S}} \bar{\alpha}(\bar{y}) \bar{\gamma}^{(0)}_{\epsilon,\bar{y}}(w, \bar{z}),$$

where $\bar{z}$ again is the *first* appropriate state in $D' \cap \bar{S}$ (since we find an interactive jump probability of zero for all other states in D'). We now apply our definition of $\bar{\gamma}$ to find

$$\sum_{\bar{y} \in D \cap \bar{S}} \bar{\alpha}(\bar{y}) \sum_{y \in D \cap S} \sum_{z \in D_3 \cap S} \frac{\alpha(y)}{\alpha(D \cap S)} \bar{\gamma}^{(0)}_{\epsilon,y}(w, z)$$
$$= \bar{\alpha}(D \cap \bar{S}) \sum_{y \in D \cap S} \sum_{z \in D' \cap S} \frac{\alpha(y)}{\alpha(D \cap S)} \bar{\gamma}^{(0)}_{\epsilon,y}(w, z)$$
$$= \sum_{y \in D \cap S} \sum_{z \in D' \cap S} \alpha(y) \bar{\gamma}^{(0)}_{\epsilon,y}(w, z),$$

because $\mathcal{E}$ relates the initial distributions of P and $\bar{P}$. The above obviously equals $\Pr(Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S)$, which shows that

$$\Pr(Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S) = \Pr(\bar{Z}^{(0)} \in D \cap \bar{S} \times \{w\} \times D' \cap \bar{S}).$$

We now consider the case $n = 0$ and $t > 0$. We have

$$\Pr(Z^{(t)} \in D \cap S \times \{w\} \times D' \cap S)$$
$$= \Pr(Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S \wedge K_1 > t)$$
$$= \Pr(K_1 > t \mid Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S) \Pr(Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S).$$

A.16

For the first factor we find, by applying Theorem 45 that

$$\begin{aligned}&\Pr(K_1 > t \mid Z^{(0)} \in D \cap S \times \{w\} \times D' \cap S)\\ &\quad = 1 - e^{-q_{D'}t} = \Pr(\bar{K}_1 > t \mid \bar{Z}^{(0)} \in D \cap \bar{S} \times \{w\} \times D' \cap \bar{S}).\end{aligned}$$

The second factor of A.16 is the case $n = 0, t = 0$, for which we already know that we find the same probability for P as for $\bar{P}$.

Now we consider the case $n > 0$ and use as our induction hypothesis that for the set of *any* set of paths over equivalence classes with $n-1$ equivalence-class jumps of the form A.13 (denoted H_{n-1}), equivalence classes D and D' of $\mathcal{E}$, and a word $w \in \mathcal{L}^V$, we have

$$\begin{aligned}&\Pr(Z^{(t)} \in H_{n-1} \times (0,\infty) \times D \cap S \times w \times D' \cap S)\\ &\quad = \Pr(\bar{Z}^{(t)} \in H_{n-1} \times (0,\infty) \times D \cap \bar{S} \times w \times D' \cap \bar{S}).\end{aligned}$$

In the following, the states x, y, z will be states in S, whereas the states $\bar{x}, \bar{y}, \bar{z}$ will be states in $\bar{S}$. For the n-th jump probability of $\bar{P}$ we now find

$$\begin{aligned}&\Pr(\bar{Z}^{(t)} \in H_n \times (0,\infty) \times D \cap \bar{S} \times w \times D' \cap \bar{S})\\ &= \int_0^\infty \int_{\substack{\bar{\sigma}\in H_n\\ \bar{\sigma}_t(n)<t\\ \bar{\sigma}_z(n)=\bar{x}\notin D}} \Pr(\bar{Z}^{(s)} \in d\bar{\sigma}) \sum_{\bar{y}\in D} \bar{q}_{\bar{x},\bar{y}} \sum_{\bar{z}\in D'} \bar{\gamma}^{(s)}_{\bar{\sigma},\bar{y}}(w,\bar{z}) e^{-\bar{q}_D(t-s)} ds\\ &= \int_0^\infty \int_{\substack{\bar{\sigma}\in H_n\\ \bar{\sigma}_t(n)<t\\ \bar{\sigma}_z(n)=\bar{x}\notin D}} \Pr(\bar{Z}^{(s)} \in d\bar{\sigma}) \sum_{\bar{y}\in D} \bar{q}_{\bar{x},\bar{y}} \bar{\gamma}^{(s)}_{\bar{\sigma},\bar{y}}(w,\bar{z}) e^{-\bar{q}_D(t-s)} ds,\end{aligned}$$

where $\bar{z}$ is again our *first* state in D' such that there is an interactive path from $\bar{y}$ to $\bar{z}$ with word w according to our arbitrary ordering of the states in $\bar{S}$. Since H_n is of the form A.13 we have that $[\bar{\sigma}]_{\mathcal{E}} = H_n$. We now fill in our definition of $\bar{\gamma}$ to find

$$\begin{aligned}&\int_0^\infty \sum_{D''\neq D} \int_{\substack{\bar{\sigma}\in[\bar{\sigma}]_{\mathcal{E}}\\ \bar{\sigma}_t(n)<t\\ \bar{\sigma}_z(n)=\bar{x}\in D''}} \Pr(\bar{Z}^{(s)} \in d\bar{\sigma}) \sum_{\bar{y}\in D} \bar{q}_{\bar{x},\bar{y}}\\ &\sum_{x\in D''} \sum_{y\in D} \sum_{z\in D'} \int_{\substack{\sigma\in[\bar{\sigma}]_{\mathcal{E}}\\ last(\sigma)=x}} \Pr(Z^{(s)} \in d\sigma \mid Z^{(s)} \in [\bar{\sigma}]_{\mathcal{E}}) \frac{q_{x,y}}{q_{x,D}} \gamma^{(s)}_{\sigma,y}(w,z) e^{-\bar{q}_D(t-s)} ds\\ &= \int_0^\infty \sum_{D''\neq D} \Pr(\bar{Z}^{(s)} \in [\bar{\sigma}]_{\mathcal{E}})\\ &\sum_{x\in D''} \sum_{y\in D} \sum_{z\in D'} \int_{\substack{\sigma\in[\bar{\sigma}]_{\mathcal{E}}\\ last(\sigma)=x}} \Pr(Z^{(s)} \in d\sigma \mid Z^{(s)} \in [\bar{\sigma}]_{\mathcal{E}}) q_{x,y} \gamma^{(s)}_{\sigma,y}(w,z) e^{-\bar{q}_D(t-s)} ds\end{aligned}$$

Now we use the fact that P and $\bar{P}$ are bisimilar and thus $\bar{q}_D = q_D$ as well as our induction hypothesis to find $\Pr(\bar{Z}^{(s)} \in [\bar{\sigma}]_{\mathcal{E}}) = \Pr(\bar{Z}^{(s)} \in H_n) = \Pr(Z^{(s)} \in H_n) = \Pr(Z^{(s)} \in [\bar{\sigma}]_{\mathcal{E}})$, which yields

$$\sum_{y\in D} \sum_{z\in D'} \int_0^\infty \int_{\substack{\sigma\in H_n\\ last(\sigma)=x\notin D}} \Pr(Z^{(s)} \in d\sigma) q_{x,y} \gamma^{(s)}_{\sigma,y}(w,z) e^{-q_D(t-s)} ds$$

which equals

$$\Pr(Z^{(t)} \in H_n \times (0,\infty) \times D \cap S \times w \times D' \cap S),$$

which completes the proof that

$$\Pr(Z^{(t)} \in H_n \times (0,\infty) \times D \times \{w\} \times D') = \Pr(\bar{Z}^{(t)} \in H_n \times (0,\infty) \times D \times \{w\} \times D').$$

for all $n \in \mathbb{N}$.

For the transient state probabilities of X_{post} we have

$$\begin{aligned}
&\Pr(X^{(t)}_{\mathsf{post}} \in D \cap S) \\
&= \sum_{n \in \mathbb{N}} \Pr(K_{n+1} > t, X^{(t)}_{\mathsf{post}} \in D \cap S) \\
&= \sum_{H_n \in upaths^{(n)}_{S/\mathcal{E},A}} \sum_{D' \in S \cup \bar{S}/\mathcal{E}} \sum_{w \in \mathcal{L}^V} \Pr(Z^{(t)} \in H_n \times (0,\infty) \times D' \times \{w\} \times D),
\end{aligned}$$

where $upaths^{(n)}_{S/\mathcal{E},A}$ is the set of all paths of length n of the form (A.13). Note that the first step only holds because we know X_{post} is regular, i.e., the probability of making infinitely many jumps before time t is zero. Now we can apply (A.14) to find that the above equals

$$\begin{aligned}
&\sum_{H_n \in upaths^{(n)}_{S/\mathcal{E},A}} \sum_{D' \in S \cup \bar{S}/\mathcal{E}} \sum_{w \in \mathcal{L}^V} \Pr(\bar{Z}^{(t)} \in H_n \times (0,\infty) \times D' \times \{w\} \times D) \\
&= \sum_{n \in \mathbb{N}} \Pr(K_{n+1} > t, \bar{X}^{(t)}_{\mathsf{post}} \in D \cap \bar{S}) \\
&= \Pr(\bar{X}^{(t)}_{\mathsf{post}} \in D \cap \bar{S}),
\end{aligned}$$

which completes the proof. □

A.2.4 Proof of Theorem 49

A stable state x in S is stochastically reachable if and only if there exists an interactive jump scheduler γ, which induces closed behaviour X, such that for all time-points $t \in \mathbb{R}_{\geq 0}$, with $t > 0$, the probability that X_{post} occupies x at time t using finitely many jumps is greater than zero. That is,

$$SR(x) \Leftrightarrow \exists \gamma \cdot \forall t > 0 \cdot \Pr(X^{(t)}_{\mathsf{post}} = x, J_\infty > t) > 0. \qquad \text{(A.17)}$$

Proof. We first prove the left-to-right implication. Since x is stochastically reachable, there exists a finite path σ from an initial state $x \in S$ of P to the stable state x. By Theorem 48 we then find, for some $n \in \mathbb{N}$, states $x_1, \ldots, x_n \in S$ and stable states $y_1, \ldots, y_n \in S_s$ such that the conditions of Theorem 48 hold. For every $1 \leq i \leq n$

we know that there is a finite interactive path from x_i to y_i. Let w_i be the sequence of visible actions along this path. We then have that $(w_i, y_i) \in \mathit{FairRT}(x_i)$. We now choose the interactive jump scheduler γ to fulfil

$$\gamma^{(t)}_{\sigma',x_i}(w_i, y_i) = 1, \qquad 1 \leq i \leq n, t \in \mathbb{R}_{\geq 0},$$

for all σ' that follow the state transitions of σ up to x_i. For $t = 0$ we pick

$$\gamma^{(0)}_{\epsilon,x_1}(w_1, y_1) = 1.$$

All other decisions of γ are chosen arbitrarily. Since (w_i, y_i) is a fair reach-trace of x_i for all indices i, we have that this γ is indeed an interactive jump scheduler for P.

For each $1 \leq i \leq n$ we will consider a set of timed paths in $\mathit{FinPaths}_{S,A}$. For $i = 1$ we consider the singleton set

$$H_1 = \{x_1\} \times \{w_1\} \times \{y_1\},$$

and for all $1 < i \leq n$ we consider the set of paths

$$H_i = H_{i-1} \times [0, \infty) \times \{x_i\} \times \{w_i\} \times \{y_i\}.$$

We will now show, by induction on i that for the behaviour X of P induced by scheduler γ (defined above) and with history process Z we have

$$\Pr(Z^{(t)} \in H_i) > 0,$$

for all $t > 0$.

For the case $i = 1$ we have

$$\begin{aligned}\Pr(Z^{(t)} \in H_1) &= \Pr(Z^{(t)} \in \{x_1\} \times \{w_1\} \times \{y_1\}) \\ &= \alpha_{x_1} \gamma^{(0)}_{\epsilon,x_1}(w_1, y_1) e^{-q_{y_1} t} \\ &= \alpha_{x_1} e^{-q_{y_1} t} > 0,\end{aligned}$$

since x_1 is an initial state and q_{y_1} is finite.

For the case $i > 1$ we use as our induction assumption that

$$\Pr(Z^{(t)} \in H_{i-1}) > 0,$$

for all $t > 0$. We now find

$$\begin{aligned}&\Pr(Z^{(t)} \in H_{i-1} \times [0, \infty)\{x_i\} \times \{w_i\} \times \{y_i\}) \\ &\quad = \int_0^\infty \int_{\sigma' \in H_{i-1}} \Pr(Z^{(s)} \in d\sigma') q_{y_{i-1},x_i} \gamma^{(s)}_{\sigma',x_i}(w_i, y_i) e^{-q_{y_i}(t-s)} ds,\end{aligned}$$

since the last state of any path in H_{i-1} is y_{i-1}. Now we have that the factors q_{y_{i-1},x_i} and $\gamma^{(s)}_{\sigma,x_i}(w_i, y_i)$ and $e^{-q_{y_i}(t-s)}$ are all greater than zero for the time-interval $0 < s < t$ and the integral

$$\int_{\sigma' \in H_{i-1}} \Pr(Z^{(s)} \in d\sigma') = \Pr(Z^{(t)} \in H_{i-1})$$

is also greater than zero by the induction assumption. It follows that $\Pr(Z^{(t)} \in H_i)$ is also greater than zero for all $t > 0$.

For the probability to be in state y_n at any time-point $t > 0$ we then have

$$\Pr(X^{(t)}_{\mathsf{post}} = y_i, J_\infty > t) > \Pr(Z^{(t)} \in H_n) > 0.$$

This completes the proof of the forward implication

We now prove te reverse implication. Given that $\Pr(X^{(t)}_{\mathsf{post}} = x, J_\infty > t) > 0$ there must be a number of jumps $n \geq 1$ such that $\Pr(X^{(t)}_{\mathsf{post}} = x, J_{n-1} > t) > 0$. We then have that there must be a set of paths

$$H_n = \{x_1\} \times \{w_1\} \times \{y_1\} \times [0, \infty) \times \ldots \times \{x_n\} \times \{w_n\} \times \{y_n\}$$

such that $y_n = x$ and $\Pr(Z^{(t)} \in H_n) > 0$, where Z is the history process of X.

We first consider the case $n > 1$. Let H_{n-1} be the set of paths consisting of the $n-1$-jump prefixes of H_n. We have

$$\begin{aligned}\Pr(Z^{(t)} \in H_n) &= \Pr(Z^{(t)} \in H_{n-1} \times [0, \infty)\{x_n\} \times \{w_n\} \times \{y_n\}) \\ &= \int_0^\infty \int_{\sigma' \in H_{n-1}} \Pr(Z^{(s)} \in d\sigma') q_{y_{n-1},x_n} \gamma^{(s)}_{\sigma',x_n}(w_n, y_n) e^{-q_{y_n}(t-s)} ds.\end{aligned}$$

It follows that q_{y_{n-1},x_n} must be greater than zero, which means y_{n-1} is stable and there is a Markovian transition from y_{n-1} to x_n. Furthermore $\gamma^{(s)}_{\sigma',x_n}(w_n, y_n)$ must be non-zero for some values of σ' and s; the consequence is that (w_n, y_n) must be a fair-reach trace of x_n. Finally, the integral $\int_{\sigma' \in H_{n-1}} \Pr(Z^{(s)} \in d\sigma')$ must be greater than zero as well, which means that $\Pr(Z^{(t')} \in H_{n-1}$ is greater than zero for some values $t' > 0$. We have found that there is a plausible path from y_{n-1} to $y_n = x$. Using the same reasoning we will find a plausible path from a state y_{n-2} to state $y_n - 1$ and so forth. In the end we will have a plausible path from a state y_1 to state $y_n = x$ and $\Pr(Z^{(t')} \in H_1) > 0$ for some $t' > 0$. We then have

$$\begin{aligned}\Pr(Z^{(t')} \in H_1) &= \Pr(Z^{(t')} \in \{x_1\} \times \{w_1\} \times \{y_1\}) \\ &= \alpha_{x_1} \gamma^{(0)}_{\epsilon,x_1}(w_1, y_1) e^{-q_{y_1} t'} > 0.\end{aligned}$$

It follows that $\gamma^{(0)}_{\epsilon,x_1}(w_1, y_1) > 0$, indicating that there is (w_1, y_1) is a fair reach-trace of x_1 and $\alpha_{x_1} > 0$, which means x_1 is an initial state. We have then shown that there is a plausible path from x_1 to y_n and x_1 is an initial state. □

A.2.5 Proof of Proposition 26

For any interactive jump scheduler γ of P, $f_E(\gamma)$ is indeed a full-history measurable scheduler of M, and $f_I(f_E(\gamma)) = \gamma$. Similarly, for any full-history measurable scheduler D of M, we have that $f_I(D)$ is indeed an interactive jump scheduler of P, and $f_E(f_I(D)) = D$.

Proof. Consider an interactive jump scheduler γ of P and its counterpart $D = f_E(\gamma)$. We will now verify that it is indeed a full-history measurable scheduler of M as per Definition 86. We now fix a path $\sigma \in \mathsf{CPaths}$ and consider the function $D(\sigma, \cdot)$. For a path of length zero $\langle x \rangle$ with $x \in S$ we have $D(\langle x \rangle, \cdot) = \gamma^{(0)}_{\epsilon,x}(\cdot)$. It then follows that this function is a probability function which assigns positive probability only to fair reach-traces of x, which are exactly the enabled actions of x. For a path σ of length $n > 0$ we have that we can find a path $\sigma' \in \mathit{FFPaths}^{(n-1)}_P$, a time-point $t \in \mathbb{R}_{\geq 0}$ and a state $x \in S$ such that $\sigma = \mathsf{EC}(\sigma', t, x)$. We then have $D(\sigma, \cdot) = \gamma^{(t)}_{\sigma',x}(\cdot)$. It then again follows that this function is a probability function which assigns positive probability only to fair reach-traces of x, which are exactly the enabled actions of x, which is the last state of σ. The measurability of D follows directly from the measurability of γ. We then have that D is indeed a full-history measurable scheduler of M.

Now consider the function $\bar{\gamma} = f_I(D)$. We must show that $\bar{\gamma} = \gamma$, i.e., that for any finite timed path $\sigma \in \mathit{FinPaths}_{S,A}$, states $x, y \in S$, time-point $t \in \mathbb{R}_{\geq 0}$, and sequence $w \in \mathcal{L}^V$ we have $\bar{\gamma}^{(t)}_{\sigma,x}(w, y) = \gamma^{(t)}_{\sigma,x}(w, y)$. This follows immediately from the fact that f_E and f_I are defined symmetrically.

In the same way we can show for a full-history measurable scheduler D of M as per Definition 86, that its counterpart $\gamma = f_I(D)$ is indeed an interactive jump scheduler for P and that $f_E(\gamma) = D$. □

A.2.6 Proof of Theorem 50

For any interactive jump scheduler γ for P, which induces a closed behaviour X with history process Z, and its counterpart $D = f_E(\gamma)$ for M, we have that

1. for a state $x \in S$ we have

$$\Pr(Z^{(J_0)} \in \{x\} \times \mathit{FairRT}(x)) = \mathcal{P}^{(0)}_D(\{x\}), \text{ and} \tag{A.18}$$

2. given a measurable set of finite fair timed paths of length $n \in \mathbb{N}_0$

$$H_n = \{(x_0, w_0, y_0)\} \times (s_1, u_1] \times \{(x_1, w_1, y_1)\} \times \ldots \times (s_n, u_n] \times \{(x_n, w_n, y_n\},$$

with states $x_0, y_0, \ldots, x_n, y_n \in S$, action-sequences $w_0, \ldots, w_n \in \mathcal{L}^V$, time-points $s_1, u_1, \ldots, s_n, u_n \in \mathbb{R}_{\geq 0}$, such that $(w_i, y_i) \in \mathit{FairRT}(x_i)$ for all $0 \leq i < n$, and $y_i \neq x_{i+1}$ for all $0 \leq i \leq n$, we have for time-points $s, u \in \mathbb{R}_{\geq 0}$, a state $x \in S \setminus \{y_n\}$, that

$$\begin{aligned} &\Pr(Z^{(J_{n+1})} \in H_n \times (s, u] \times \{x\} \times \mathit{FairRT}(x)) = \\ &\quad \mathcal{P}^{(n+1)}_D(\mathsf{EC}(H_n \times (s, u] \times \{x\} \times \mathit{FairRT}(x))). \end{aligned} \tag{A.19}$$

Proof. We first prove the first part of Theorem 50. For the left-hand side of (A.18) we find,

$$\Pr(Z^{(J_0)} \in \{x\} \times \mathit{FairRT}(x)) = \alpha(x) \sum_{\langle w,y \rangle \in \mathit{FairRT}(x)} \gamma^{(0)}_{\epsilon,x}(w, y) \qquad = \alpha(x),$$

since $\gamma^{(0)}_{\epsilon,x}$ is a probability function which assigns positive probability only to reach-traces in $FairRT(x)$. Since α is a Dirac distribution we have that the above equals

$$\begin{cases} 1, & \text{if } x = \hat{x}, \\ 0, & \text{otherwise.} \end{cases}$$

For the right-hand side of (A.19) we also find

$$\mathcal{P}^{(1)}_D(\{(x)\}) = \begin{cases} 1, & \text{if } x = \hat{x}, \\ 0, & \text{otherwise.} \end{cases}$$

This means (A.18) holds for all $x \in S$.

We prove the second part of Theorem 50 by induction on the path length n. As our induction assumption we assume that,

$$\begin{aligned} &\Pr(Z^{(J_n)} \in H'_{n-1} \times (s', u'] \times \{x'\} \times FairRT(x')) = \\ &\quad \mathcal{P}^{(n)}_D(\mathsf{EC}(H'_{n-1} \times (s', u'] \times \{x'\} \times FairRT(x'))), \end{aligned}$$

for a measurable set of finite fair timed paths H'_{n-1} of length $n-1$ of P constructed in the same way as the set H_n in Theorem 50.

$$\begin{aligned} &\Pr(Z^{(J_{n+1})} \in H_n \times (s, u] \times \{x\} \times FairRT(x)) \\ &= \sum_{\langle w,y\rangle \in FairRT(x)} \int_s^u \int_{\substack{\sigma \in H_n \\ \sigma_t(n) > t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-q_{y_n}(t-\sigma_t(n))} q_{y_n,x} \gamma^{(t)}_{\sigma,x}(w,y) dt \\ &= \int_s^u \int_{\substack{\sigma \in H_n \\ \sigma_t(n) > t}} \Pr(Z^{(J_n)} \in d\sigma) e^{-q_{y_n}(t-\sigma_t(n))} q_{y_n,x} dt, \end{aligned}$$

since $\gamma^{(t)}_{\sigma,x}(\cdot)$ is a probability function which assigns strictly positive probability only to fair reach-traces of x. Let σ' be the first part of σ, i.e., $\sigma = \sigma' \circ (\sigma_t(n), x_n, w_n, y_n$. Furthermore, let the measurable set $d\sigma''$ be defined as

$$\lim_{h \to 0} d\sigma' \times (\sigma_t(n) + h] \times \{x_n\} \times FairRT(x_n),$$

i.e., $d\sigma''$ is the same set of paths as $d\sigma$, except that all fair-reach traces of x_n are considered instead of just (w_n, y_n). We then have that the above equals

$$\begin{aligned} &\int_s^u \int_{\substack{\sigma \in H_n \\ \sigma_t(n) > t}} \Pr(Z^{(J_n)} \in d\sigma'') e^{-q_{y_n}(t-\sigma_t(n))} q_{y_n,x} \gamma^{(\sigma_t(n))}_{\sigma',x_n}(w_n, y_n) dt \\ &= \int_s^u \int_{\substack{\sigma \in H_n \\ \sigma_t(n) > t}} \mathcal{P}^{(n)}_D(\mathsf{EC}(d\sigma'')) e^{-R(x_n,\langle w_n,y_n\rangle)(t-\sigma_t(n))} \\ &\qquad \cdot R(x_n, \langle w_n, y_n\rangle, x) D(\mathsf{EC}(\sigma), \langle w_n, y_n\rangle) dt \\ &= \mathcal{P}^{(n+1)}_D(\mathsf{EC}(H_n \times (s, u] \times \{x\} \times FairRT(x)), \end{aligned}$$

where we applied the induction assumption, (7.8), Definition 92, and Definition 87. □

A.2.7 Proof of Theorem 51

Consider a closed I/O-IMC $P = (S, A, R^I, R^M, \alpha)$ with no internal transitions and where for each state $x \in S$ we have

$$x \xrightarrow{a} y, x \xrightarrow{b} z \text{ implies } a = b, y = z.$$

For any closed behaviour X of P that is non-explosive, i.e.,

$$\Pr(J_\infty = \infty) = 1,$$

we have that X_{post} is a Markov chain.

Proof. We will show that X_{post} satisfies the Markov condition "up to $o(h)$" (cf. 3.37). For distinct stable states $x, y \in S$, time-points $t < t + h \in \mathbb{R}_{\geq 0}$, and a finite timed path σ ending in state x we have

$$\begin{aligned}
&\Pr(X_{\mathsf{post}}^{(t+h)} = y \mid X_{\mathsf{post}}^{(t)} = x, Z^{(t)} = \sigma) \\
&= \sum_{w \in \mathcal{L}^V} \sum_{z \in S} \Pr(X_{\mathsf{post}}^{(J_1^{(t)})} = y, W^{(J_1^{(t)})} = w, X_{\mathsf{pre}}^{(J_1^{(t)})} = z, J_1^{(t)} \leq t + h \\
&\qquad\qquad \mid X_{\mathsf{post}}^{(t)} = x, Z^{(t)} = \sigma) + o(h),
\end{aligned}$$

since the probability of two jumps occurring is $o(h)$. We split the above probability to find that it equals

$$\begin{aligned}
&\sum_{w \in \mathcal{L}^V} \sum_{z \in S} \\
&\Pr(X_{\mathsf{post}}^{(J_1^{(t)})} = y, W^{(J_1^{(t)})} = w \mid X_{\mathsf{pre}}^{(J_1^{(t)})} = z, J_1^{(t)} \leq t + h, X_{\mathsf{post}}^{(t)} = x, Z^{(t)} = \sigma) \\
&\quad \cdot \Pr(X_{\mathsf{pre}}^{(J_1^{(t)})} = z, J_1^{(t)} \leq t + h \mid X_{\mathsf{post}}^{(t)} = x, Z^{(t)} = \sigma) + o(h) \\
&= \sum_{w \in \mathcal{L}^V} \sum_{z \in S} \gamma_{\sigma,z}^{(t)}(w, y) q_{x,z} h + o(h).
\end{aligned}$$

Recall that each state z has only a single non-divergent fair reach-trace. Since $\gamma_{\sigma,z}^{(t)}$ is a probability function it must assign one to this reach-trace and zero to all others. We then find that the above equals

$$\sum_{w \in \mathcal{L}^V} \sum_{\substack{z \in S \\ (w,y) \in FairRT(z)}} q_{x,z} h + o(h).$$

This shows that 3.37 holds and we can then apply Theorem 2 to show that X_{post} is indeed a Markov chain. □

Bibliography

[1] W. J. Anderson. *Continuous Time Markov Chains: An Applications-Oriented Approach*. Springer-Verlag, 1991.

[2] H. C. Bohnenkamp and B. R. Haverkort. Semi-numerical solution of stochastic process algebra models. In *Proceedings of the 5th International AMAST Workshop*, pages 228–243, 1999.

[3] H. Boudali, P. Crouzen, B. R. Haverkort, M. Kuntz, and M. Stoelinga. Architectural dependability evaluation with Arcade. In *Proceedings of the 38th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 512–521, 2008.

[4] H. Boudali, P. Crouzen, and M. Stoelinga. A compositional semantics for dynamic fault trees in terms of interactive Markov chains. In *Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, pages 441–456, 2007.

[5] H. Boudali, P. Crouzen, and M. Stoelinga. Coral – a tool for compositional reliability and availability analysis. In *ARTIST workshop: Tool Platforms for embedded systems modelling, analysis and validation*, 2007.

[6] H. Boudali, P. Crouzen, and M. Stoelinga. Dynamic fault tree analysis using input/output interactive Markov chains. In *Proceedings of the 37th IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 708–717, June 2007.

[7] H. Boudali, P. Crouzen, and M. Stoelinga. A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE Transactions on Dependable and Secure Computing*, 7(2):128–143, 2010.

[8] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31(1):59–75, March 1994.

[9] P. Buchholz, E. M. Hahn, H. Hermanns, and L. Zhang. Model checking algorithms for CTMDPs. In *CAV*, pages 225–242, 2011.

[10] P. Buchholz and I. Schulz. Numerical analysis of continuous time Markov decision processes over finite horizons. *Computers & OR*, 38(3):651–659, 2011.

[11] L. Cheung, N. A. Lynch, R. Segala, and F. W. Vaandrager. Switched PIOA: Parallel composition via distributed scheduling. *Theoretical Computer Science*, 365(1-2):83–108, 2006.

[12] T. H. Cormen. *Introduction to algorithms*. MIT Press, 2001.

[13] P. Crouzen and H. Hermanns. Aggregation ordering for massively compositional models. In *Proceedings of the 11th International Conference on Application of Concurrency to System Design (ACSD)*, pages 171–180, 2010.

[14] P. Crouzen and F. Lang. Smart reduction. In *Proceedings of the 14th International Conference on Fundamental Approaches to Software Engineering (FASE)*, pages 111–126, 2011.

[15] F. Didier, T. A. Henzinger, M. Mateescu, and V. Wolf. Approximation of event probabilities in noisy cellular processes. In *Proceedings of the 7th International Conference on Computational Methods in Systems Biology (CMSB)*, pages 173–188, 2009.

[16] J. L. Doob. *Stochastic Processes*. Wiley, 1990.

[17] D. Freedman. *Markov chains*. Springer-Verlag, 1971.

[18] H. Garavel, F. Lang, R. Mateescu, and W. Serwe. CADP 2010: A toolbox for the construction and analysis of distributed processes. In *Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2011.

[19] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[20] S. Giro and P. R. D'Argenio. On the expressive power of schedulers in distributed probabilistic systems. *Electronic Notes on Theoretical Computer Science*, 253(3):45–71, 2009.

[21] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. Infamy: An infinite-state Markov model checker. In *Proceedings of the 21st International Conference on Computer Aided Verification (CAV)*, pages 641–647, 2009.

[22] B. R. Haverkort. Markovian models for performance and dependability evaluation. In *European Educational Forum: School on Formal Methods and Performance Analysis*, pages 38–83, 2000.

[23] H. Hermanns. *Interactive Markov Chains*. Springer, 1994.

[24] H. Hermanns. *Interactive Markov Chains*, volume 2428 of *Lecture Notes in Computer Science*. Heidelberg: Springer Berlin, 2002.

[25] H. Hermanns and J.-P. Katoen. Automated compositional Markov chain generation for a plain-old telephone system. *Science of Computer Programming*, 36(1):97–127, 2000.

[26] H. Hermanns and L. Zhang. From concurrency models to numbers - performance and dependability. In *Software and Systems Safety - Specification and Verification*, pages 182–210. 2011.

[27] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[28] R. A. Howard. *Dynamic Probabilistic Systems.* John Wiley and Sons, 1971.

[29] A. Jensen. Markov chains as an aid in the study of Markov processes. *Skand. Aktuarietidskrift*, 3:87–91, 1953.

[30] S. Johr. *Model Checking Compositional Markov Systems*. PhD thesis, Saarland University, Germany, 2007.

[31] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen. The ins and outs of the probabilistic model checker MRMC. In *Proceedings of the 6th International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 167–176, 2009.

[32] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV)*, pages 585–591, 2011.

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

[34] S. Maaß. Translating Arcade models into Modest code. Bachelor's thesis, Saarland University, 2010.

[35] R. Milner. *Communication and Concurrency*. Prentice Hall, 1984.

[36] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of chemical physics*, 124:44–104, 2006.

[37] M. R. Neuhäußer, M. Stoelinga, and J.-P. Katoen. Delayed nondeterminism in continuous-time Markov decision processes. In *Proceedings of the 7th international conference on formal modeling and analysis of timed systems (FOSSACS)*, pages 364–379, 2009.

[38] M. R. Neuhäußer and L. Zhang. Time-bounded reachability probabilities in continuous-time Markov decision processes. In *Proceedings of the 7th international conference on the quantitative evaluation of systems (QEST)*, 2010.

[39] X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In *REX Workshop*, pages 526–548, 1991.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[41] L. C. Paulson. *Logic and computation: interactive proof with Cambridge LCF*, volume 2. Cambridge University Press, 1990.

[42] R. Pulungan. *Reduction of Acyclic Phase-Type Representations*. PhD thesis, Saarland University, 2009.

[43] A. Remke and B. R. Haverkort. CSL model checking algorithms for infinite-state structured Markov chains. In *Formal Modeling and Analysis of Timed Systems*, pages 336–351. Springer, 2007.

[44] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.

[45] R. B. Sidje, K. Burrage, and S. MacNamara. Inexact uniformization method for computing transient distributions of Markov chains. *SIAM Journal on Scientific Computing*, 29(6):2562–2580, 2007.

[46] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[47] T. A. Sudkamp. *Languages and machines: an introduction to the theory of computer science*. Addison-Wesley, 1997.

[48] J. Tretmans. Test generation with inputs, outputs, and quiescence. In *Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, pages 127–146, 1996.

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

[50] R. Wimmer, B. Braitling, B. Becker, E. M. Hahn, P. Crouzen, H. Hermanns, A. Dhama, and O. E. Theel. Symblicit calculation of long-run averages for concurrent probabilistic systems. In *Proceedings of the 7th International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 27–36, 2010.

[51] V. Wolf. *Equivalences on phasetype processes*. PhD thesis, University of Mannheim, 2008.

[52] N. Wolovick and S. Johr. A characterization of meaningful schedulers for continuous-time Markov decision processes. In *Proceedings of the 4th international conference on formal modeling and analysis of timed systems (FORMATS)*, volume 2402 of *LNCS*, pages 352–367, 2006.

[53] S.-H. Wu, S. A. Smolka, and E. W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176(1-2):1–38, 1997.