Coalgebraic Analysis of Probabilistic Systems

Coalgebraic Analysis of Probabilistic Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 3 november 2005 om 16.00 uur

door

Ana Sokolova

geboren te Skopje, Macedonië

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. J.C.M. Baeten

Copromotor:

dr. E.P. de Vink

IPA Dissertation Series 2005-16

Typeset using LaTeX

Printed by University Press Facilities, Eindhoven

Cover design by Maja Sokolova, adaptation by Jan-Willem Luiten

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Sokolova, Ana

Coalgebraic analysis of probabilistic systems / Ana Sokolova. Eindhoven : Technische Universiteit Eindhoven, 2005.

Proefschrift. ISBN 90-386-0684-2

NUR 993

Subject headings : stochastic models / category theory

CR Subject Classification (1998): F.1.1, F.1.2, F.3.2

The work on this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics). The author was employed at the Eindhoven University of Technology, supported by PROGRESS / STW (Dutch funding agency for university research), project EES.5202, (A)MaPAoTS.

Contents

Preface

As I write these lines, a period of over four great years that led to this thesis ends. The TU/e is a remarkably vivid place and it provides a nice environment for (foreign) PhD students. The Netherlands is a beautiful country. I could fill a whole thesis on the life in the past four years and all the people that contributed to my life and to my work. Here, I can only mention some of them.

First of all I would like to thank my promotor, Jos Baeten, for offering me a PhD position in the beginning of 2001. Jos also deserves the credit for the friendly and relaxed atmosphere in the Formal Methods Group and the regular group lunches. He has all my respect as a scientist and a person. All these years I had a very pleasant feeling that Jos believes in me.

I am happy to have Erik de Vink as a supervisor and copromotor. I learned a lot from Erik and I very much appreciate that he could handle my sometimes stubborn reactions. Often I would realize with a delay that he was right when pointing to a trap in a proof or to a shortcut. Erik is also a coauthor of several papers on which this thesis is based. I greatly appreciated Jos' and Erik's liberal attitude towards working place, which made it possible for me to often work at home, even when "home" was more than thousand kilometers away from my office.

For the valuable remarks that improved the quality of the thesis I thank the members of the reading committee: Twan Basten, Holger Hermanns and Jan Rutten. I thank Bart Jacobs, Smile Markovski and Jeroen Voeten for accepting to take part in the defense committee. Ichiro Hasuo and Bart Jacobs showed interest in my work and commented parts of the thesis, for which I am grateful.

The thesis is based on several papers which are joint work with Falk Bartels, Erik de Vink and Harald Woracek. I am grateful to my coauthors for their contributions and for making the work so enjoyable. I thank the anonymous referees for their comments. Interested people always ready to discuss every small lemma make the best working environment that I can imagine. This is exactly why I also happily accepted a postdoc position in the group of Bart Jacobs, despite the fact that the office is very far from home. I thank Bart for offering me the chance of working in his group.

My work at the TU/e was supported by the PROGRESS project (A)MaPAoTS.

I appreciated the occasional project meetings and the discussions with the project members Dimitri Chkliaev, Yusuf Pribadi, Bart Theelen, Simona Vlad and Jeroen Voeten.

My research on the topics reported in this thesis was initiated by the VOSS Dagstuhl Seminar in which I took part in 2002. I thank the organizers of this seminar. In particular, I thank Holger Hermanns for supervising my early work. The discussions with him were a great source of motivation. Participation in conferences and schools was always very enjoyable and motivating. I would like to thank all the people that were a great company during conferences and summer schools. I thank IPA, and especially Tijn Borghuis, for the nice events that they organized.

I thank my (former) colleagues from the (broader) FM group Bas, Cas, Desiree, Dimitri, Elize, Erik, Erik, Francien, Georgi, Gia, Harm, Hugo, Jasen, Jerry, Jos, Kalok, Kees, Louis, Michael, Michiel, Nikola, Rob, Roel, Ronald, Ruurd, Simona, Sjouke, Sonja, Suzana, Tijn, Tim, Tim, and Uzma for the enjoyable lunches, long coffee breaks, numerous social activities like borrels, Sinterklaas events and Christmas tapas. It was nice to share the office with Dimitri and even nicer with Uzma. With Georgi, Tim, Nikola and Jasen it was common to just approach the board and start discussing whatever current problem I had. I also often enjoyed listening to their problems. I am grateful to Hugo and Erik for their help on the Dutch summary.

In the first two years in Eindhoven I spent two days per week in the Embedded Systems Institute (EESI). There I enjoyed the nice working conditions and the company of Fei, Giovanni, Harm, Jinfeng, Kalok, Luis, Michel, Mohammad, Pieter and Stephen.

I thank my close friends Suzana and Dragan for the nice time spent together and the incredible amount of help I received from them throughout these years. I can not imagine life in Eindhoven without my best friend there, Georgi. I often said that Suzana, Dragan and Georgi are my family in Eindhoven. This family increased couple of years ago when Nikola and Marija came to Eindhoven. I also thank Bruno & Julia, Caci & Bube, Cristina, Dragan, Dragan, Falk, Gabi & Katharina, Giovanni, Jasen, Kalok, Lira, Marcella & Tim, MartijnO, Mohammad, Nikola & Buba, Nevenka, Nino, Natasha & Gregor, Olga, Orce & Roberta, Roland, Simona & Bas, Uzma, Victor and Žarko for being my friends and for the nice times together in Eindhoven and (around) Vienna. In the last year, despite writing the thesis, I made another big achievement: I stopped smoking. I thank Marcella, Georgi and Tim for pushing me a bit towards this decision. Bas, Erik and Harald are thanked for their extensive support on this issue.

At the Institute of Informatics in Skopje, Macedonia, I spent many valuable years. First as an undergraduate student, later as a graduate student and a teaching assistant. I thank all my colleagues and friends from II, but especially Bibi, Lidija, Smile, Čupona and Šunka. Going further back, I wish to mention

my great math teacher and friend, Frosina. My friends from Macedonia and all over the world stayed in touch with me despite of the distance. I mention only few: Aleksandar & Maja, Ana, Ana & Tome, Anastas & Ema, Angel, Boro, Darko & Sonja, Emilija, Hajdi, Jana & Petar, Nane, Nikola & Verče, Riste, Suzi, Verica, Viki and Vlado. I thank my yoga teacher Vlado and our guru Mataji for teaching me valuable things.

The family Titz and my twin sister Maja helped me with various issues of settling in Eindhoven. The home of Maria and Rudiger felt sometimes as a replacement for my parents' home and was a nice place to meet Alexander, Maja and Nora.

I am grateful to my mother in law, Traude Woračeк, for always making me feel welcome and at home. In this she is joined by aunt Gerti. The rest of my family, my parents Pančе and Milka, my sister Tanja, my twin-sister Maja and my nieces, I thank for being always there for me, loving me and believing in me. Maja brings lot of beauty in my life. She also designed the cover for the thesis. From my parents I inherited the love for science, books and beauty. I still often say "home" thinking of the wonderful home that they made and kept for all of us.

Finally, the greatest thanks I give to Harald. For all the love that he brings in my life.

Vienna, September 2005

1

Introduction

This thesis connects and contributes to two areas of research in theoretical computer science. The first area is that of probabilistic verification, in particular probabilistic modelling and behavioral relations on probabilistic systems [GSST90, Han91, SL94, Bai98, Sto02a]. The second one is the theory of coalgebras, in particular coalgebraic modelling [Rut96, JR96, Gum99, Rut00].

These two areas of theoretical computer science are linked by the notion of a *transition system*. A transition system is an abstract object consisting of states and transitions between the states. In the area of probabilistic modelling various transition systems decorated with probabilistic information are present and are used for modelling purposes. The theory of coalgebras, closely related to category theory, provides a general approach of modelling transition systems and data structures as coalgebras of a behavior functor. When changing the behavior functor, one obtains various concrete types of systems like labelled transition systems, lists, but also most of the probabilistic system types that appear in the literature on probabilistic modelling [VR99, Mos99, BSV04].

The treated research questions in the thesis are of three kinds:

1. Questions that origin in the area of probabilistic modelling.
2. Correspondence questions: How do general coalgebraic notions instantiate in concrete systems? How do concrete notions generalize to coalgebras ?
3. Open problems in the theory of coalgebras.

The work related to the questions of type 1 started as an overview study of probabilistic models, initiated by the organizers of the VOSS Dagstuhl Seminar on Validation of Stochastic Systems [BHH$^+$04]. We address the following questions of type 1:

- What type of probabilistic models exist?
- How expressive are these models?
- Which notions of bisimulation exist for the models ?

- How do they compare in expressive power?
- How can they be composed in parallel, do they satisfy closure properties?

For these questions we conduct a detailed comparative study of the various probabilistic transition systems. The comparison of the expressive power of probabilistic systems is a central topic in the thesis. For the comparison we apply the theory of coalgebras. We use coalgebras as a unified framework that allows more abstract treatment and more general and elegant results and proofs.

Our comparison criterion is expressed in terms of maps that preserve and reflect bisimilarity. Hence, bisimulations and bisimilarity are central notions in the thesis. On the one hand, most of the probabilistic models come equipped with a notion of probabilistic bisimulation [LS91]. On the other hand, a great contribution of the theory of coalgebras is a generic notion of bisimulation [AM89]. In order to be able to interchangeably use probabilistic and coalgebraic bisimulation we address the following correspondence question.

- What is the relation between the coalgebraic bisimulation and the concrete probabilistic notions of bisimulation ?

This question is a typical question of type 2. The answer is that the coalgebraic notion always coincides with the concrete notion, when instantiated to a given type of systems. This allows us to also define bisimulations (by instantiating the coalgebraic definition) for some types of systems for which concrete notion of bisimulation was not given. Some other questions of type 2 that we discuss are:

- How do the coalgebraic and the concrete notions of simulation relate?
- How can one generalize the notion of weak bisimulation known for some concrete systems to coalgebras?

Actually, the last question is also of type 3. Defining weak bisimulation is an open problem in the theory of coalgebras. Questions of type 3 that we focus on are:

- How to define weak bisimulation for coalgebras?
- How to compose coalgebras and define paths (or other notions of linear behavior) in coalgebras?

We attack these questions using intuition and knowledge from concrete systems, in particular from probabilistic systems, as important leading examples. Having provided a (partial) solution to the problem of weak bisimulation for coalgebras, it induces a new correspondence question:

- What is the relation between the coalgebraic weak bisimulation and the concrete (probabilistic) notions of weak bisimulation?

In the remainder of the present chapter we introduce in more detail the process of modelling using transition systems and the addition of probabilities, leading to probabilistic models. Moreover, we make the step to coalgebras and coalgebraic modelling as higher-level representation of transition systems. Having introduced the framework, we discuss in more detail the research objectives and the contributions of the thesis. We end the introduction with a brief overview of related work.

Let us point out that an important goal of this thesis is connecting coalgebras and probabilistic systems. Therefore, we do not assume deep knowledge in any of the mentioned areas. Notions used from category theory, coalgebra and probabilistic systems will be introduced when required.

1.1 Modelling

The aim of formal methods is to develop theories, techniques and tools for formal and preferably automatic verification of real systems. The real system can be, for example, a nuclear plant, a complicated machine, a communication protocol, or a piece of code. In any case the goal is to verify that the system has a certain property, for example, not all of the airplane motors will stop working at the same time, the traffic light will eventually turn green, or the coffee-machine returns change and delivers coffee. Actually, it is not the case that the real system is verified, except via testing. More commonly a formal model of the real system is designed and then the model is verified with respect to a given property. The model is a representation of the system in some formalism, an abstraction of the real system. The level of abstraction of the model with respect to the real system depends on the expressive power of the formalism and the opinion of the model designer regarding the features of the real system that are relevant for the property that is to be verified.

Hence, the model is verified with respect to some property and not the real system itself. However, having verified a property of the model brings confidence in the design of the real system. More importantly, if a property is not verified, i.e. an erroneous behavior is found in the model, then this suggests an error in the real system itself or in the process of modelling.

Therefore the models, the modelling formalisms and their expressive power are important issues in formal methods. There are various formalisms in which a model of a concurrent system can be expressed such as transition systems (automata, state based models) [Mil89, BK85, Rut00], process algebra [Mil89, Hoa85, BK85, BW90], Markov chains [KS76, How71] and others.

In this thesis, we focus on models of systems that are transition systems. We use the terms model, (transition) system, state machine and automaton inter-

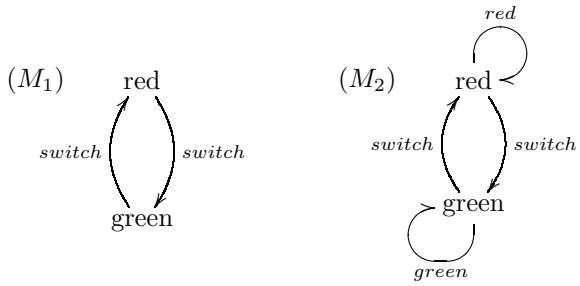
changeably as synonyms.

Transition systems

Transition systems consist of a set of states and some type of a transition relation or function. In general, we consider the system as a black box and the states as not observable for an outside observer.

For example, a simple type of transition systems are the *fully deterministic labelled transition systems*, which are pairs $\langle S, \rightarrow \rangle$ where $\rightarrow$ is a transition function from the set of states S to the set $A \times S$ for a set of action labels A. The set S contains the states of the system, and the function $\rightarrow$ describes the unique transition out of a state. The transitions are labelled by elements of the set A. We write $s \xrightarrow{a} t$ for $\rightarrow (s) = \langle a, t \rangle$, representing the transition in the system from the state s to the state t with a label a. The labels are observable from the outside. They stand for the execution of an action.

Consider the transition system (M_1) in the left diagram below.



It is a model of a simple traffic light, with two states, $S = \{\text{red}, \text{green}\}$, only one observable action *switch* and two transitions. Since the states are not observable, although we have given them names, there is not much to be observed in this model.

A more sophisticated type of transition systems, and the most commonly used one is the *labelled transition system*. A labelled transition system, or LTS, is a pair $\langle S, \rightarrow \rangle$ where S is a set of states and $\rightarrow \subseteq S \times A \times S$ is the labelled transition relation for a set of labels A. A transition $\langle s, a, t \rangle \in \rightarrow$ is denoted by $s \xrightarrow{a} t$ and is interpreted as: a transition labelled by a can be made from the state s to the state t.

We may now refine our traffic light model, so that it contains action labels as (M_2) in the right diagram above. We are aware that neither the initial model (M_1) nor the refined (M_2) are desired models of traffic lights. For example, the traffic light from (M_2) may go on performing the action *red* forever; it may also change from the state red to the state red (via two *switch* actions) without performing a single *green* action. These issues are not our concern at

the moment.

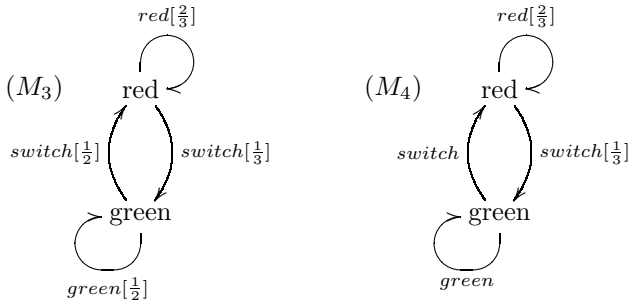
A useful feature of the labelled transition systems is the possibility to express non-determinism since $\rightarrow$ is a transition relation. So, there may be more transitions (perhaps labelled with the same label) going out from a state.

Modelling probabilistic behavior

Non-determinism is valuable for expressing incomplete knowledge about the system that is modelled, or about the behavior of the environment. However, it is often known that certain choices in a system are governed by probability distributions, or are consequences of random events. Hence, there is quantified information about the actual probability that should find its way into the models. There are various ways to enrich the labelled transition systems with probabilistic choices (e.g. [GSST90, Han91, SL94]) and these models are one of our main topics of interest. Let us look at an example of how probabilistic information can be incorporated in the model of the traffic light.

Assume first that we have a traffic light behaving as the model in our LTS version (M_2), for which we moreover know that the choice in the red state between the actions *red* and *switch* is not unknown but is governed by a probability distribution that assigns probability 2/3 to the action *red* and 1/3 to the action *switch*. Hence, in state red, action *red* will be executed (observed) in two thirds of the cases, and action *switch* in the remaining one third. Similarly, a uniform probabilistic distribution determines the choice between the actions *green* and *switch* in the state green, and we obtain the model as in the left diagram below, referred to as (M_3).

It could also be the case that we have no information at all of any probabilistic pattern in the choices in the green state. Hence we obtain a model that contains both probabilistic and non-deterministic choices, as shown in the right diagram below, called (M_4).



Various modelling choices lead to various types of systems. In the literature there are many types of transition systems with or without action labels, probabilities and non-determinism.

The type of systems corresponding to the model (M_3) in the diagram above is known as *generative probabilistic systems* [GSST90, GSS95]. They are defined as pairs $\langle S, P : S \times A \times S \to [0,1] \rangle$ of a set of states S and a probabilistic transition relation P. We write $s \xrightarrow{a[p]} t$ if $P(s,a,t) = p$ and say that a transition from s to t with label a happens with probability p. There is an extra requirement that the sum of the outgoing probabilities, if any, of every state in a generative system equals one.

Coalgebraic modelling

As we have seen, the labelled transition systems are pairs $\langle S, \to \subseteq S \times A \times S \rangle$ and the generative probabilistic systems are pairs $\langle S, P : S \times A \times S \to [0,1] \rangle$. However, usually when working with these systems we do not treat the incoming and the outgoing transitions with the same interest, we rather think of a state together with its outgoing transitions. We may equivalently represent these systems via a *transition function*, instead of transition relation, as is already the case for the deterministic systems. The labelled transition systems are pairs

$$\langle S, \alpha : S \to \mathcal{P}(A \times S) \rangle$$

using $\mathcal{P}(A \times S)$ to denote the power set of the set $A \times S$ and $s \xrightarrow{a} t \iff \langle a, t \rangle \in \alpha(s)$. The generative probabilistic systems can also be defined via a transition function, as pairs

$$\langle S, \alpha : S \to \mathcal{D}(A \times S) \rangle$$

where $\alpha(s)(\langle a, t \rangle) = P(s,a,t)$ and $\mathcal{D}(X)$, for a set X, denotes the set of all discrete probability distributions on X. This way of presenting transition systems generalizes to the notion of a *coalgebra of a functor*.

A coalgebra of a functor (or type) $\mathcal{F}$ is a pair

$$\langle S, \alpha : S \to \mathcal{F}S \rangle$$

where $\mathcal{F}S$ denotes a set, constructed in a systematic way from the set S. The set-theoretic construction $\mathcal{F}$ is described by the notion of a *functor* from category theory [Mac71, Bor94]. In a coalgebra $\langle S, \alpha \rangle$ of type $\mathcal{F}$, the set S is the carrier, or the set of states, and the map α is the transition function, or the structure map. This approach led to a general study of universal coalgebra [Rut00] as an abstract theory of transition systems and data structures. For every functor, the coalgebra approach also provides a natural notion of *homomorphism*, i.e. behavior preserving function between systems, as well as an induced notion of *coalgebraic bisimilarity* and *behavior equivalence*.

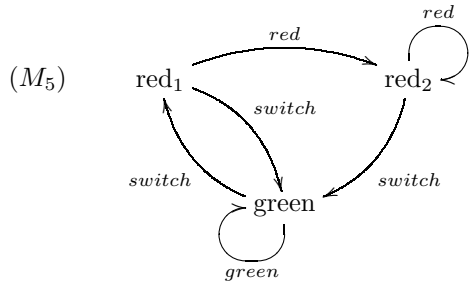
1.2 Behavior relations

The behavior of a system, or more precisely the behavior that a state in a system exhibits, is a relative notion depending on a given semantic or behavior

relation, often a semantic equivalence. Assume we have a notion of a semantic equivalence $\sim$ and we can show that the states s and t in some system are related, i.e. $s \sim t$. Then we say that s and t have the same behavior, with respect to the $\sim$ behavior relation. In the study of concurrent systems several semantic relations have been established (c.f. [Gla90, Gla93, Gla01]). Strong bisimilarity and weak bisimilarity are examples of branching-time semantics; trace equivalence is an example of a linear-time semantics.

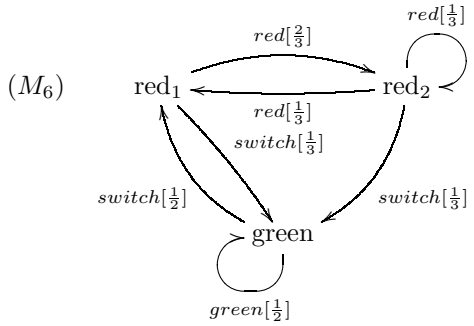
Bisimulation

Bisimulation and bisimilarity are central notions in this thesis. Bisimulation [Par81, Mil89] is a relation on the state set of a system that relates states with the same stepwise behavior. This is to say that whenever two states are related by a bisimulation relation, each step of any of them can be mimicked by a matching step of the other one. Consider the labelled transition system (M_5) from the next diagram.



The equivalence relation that puts the states red$_1$ and red$_2$ in one class, and the state green in another is a bisimulation relation since all the steps from red$_1$ can be mimicked by matching steps from red$_2$, and vice versa. For example, the transition labelled by *red* from the state red$_1$ can be mimicked by the transition labelled by *red* from the state red$_2$, and they lead to related states. Two states are bisimilar if they are related by some bisimulation. Bisimilarity is an equivalence for LTSs.

For the various probabilistic transition systems there are also notions of bisimulation and bisimilarity [LS91]. Only now, the notion of a "step" or a "transition" is different. A bisimulation relation should compare the probability to reach an equivalence class and not the probability to reach a single state. The probability of reaching a set of states is the sum of the probabilities of reaching its elements. For example, the states red$_1$ and red$_2$ in the following generative probabilistic system (M_6) are bisimilar.



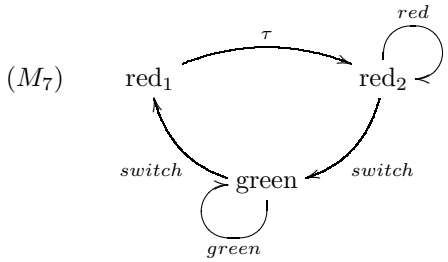
The probability of reaching the class of red states from any of the red states via the label *red* is 2/3, and the probability of reaching the class of the green state from any of the red states via the label *switch* is 1/3.

For coalgebras of a functor there is also the notion of coalgebraic bisimulation and bisimilarity [AM89]. This notion is expressed in general terms of coalgebra homomorphisms. Bisimilarity is an equivalence for a rich class of well-behaved functors. Moreover, for coalgebras there exists another notion of behavioral equivalence based on cocongruences [Kur00, Wor00]. For the well behaved functors these two notions coincide, and for the functors that are not well behaved the behavioral equivalence is the preferred notion. For example, behavioral equivalence is always an equivalence. We use cocongruences and behavioral equivalence for our comparison result.

Weak bisimulation

Weak bisimulation abstracts away from invisible behavior. It leads to weak bisimilarity which is a weaker equivalence than strong bisimilarity. Assume that in the set of labels A there is a label τ that can not be observed. Such labels may occur, for example, as a result of communication between systems [Mil89]. A relation is a weak bisimulation if whenever two states are related, then any step from each of them can be mimicked by a *weak step* from the other one. A weak step for a label a, in an LTS, is obtained by a sequence of steps whose labels form a word with visible content a, notation $\stackrel{a}{\Rightarrow}$. Consider, for example, the labelled transition systems (M_7) in the following diagram.

The states red$_1$ and red$_2$ are indistinguishable in a weak bisimulation semantics. The step red$_2 \xrightarrow{red}$ red$_2$ is mimicked by the weak step red$_1 \stackrel{red}{\Longrightarrow}$ red$_2$, i.e. red$_1 \xrightarrow{\tau}$ red$_2 \xrightarrow{red}$ red$_2$. Similar observations hold for the step with label *switch* from red$_2$. The step with label τ from red$_1$ is mimicked by an *empty* step from red$_2$.



Interestingly, there is a simple way to transform an LTS $\langle S, \rightarrow \rangle$ to an LTS $\langle S, \Rightarrow \rangle$ such that weak bisimilarity on the original system $\langle S, \rightarrow \rangle$ corresponds to strong bisimilarity on the transformed system $\langle S, \Rightarrow \rangle$.

Notions of weak bisimulation and weak bisimilarity exist for several types of probabilistic systems [Seg95b, BH97, PLS00]. Their definitions are rather involved. Still, they reflect an analogy to the weak bisimulation definition for LTSs.

On the other hand, it is not easy to define weak bisimulation for coalgebras [Rut99, Rot02, RM02]. The problem breaks down into several issues: the notion of visible vs. invisible behavior for general coalgebras is not obvious, the definition of weak steps should include composition of steps which is difficult in general. From the approach that works for LTSs, borrowing some aspects of the definition of weak bisimulation for probabilistic systems, we obtain a partial solution of the weak bisimulation problem for coalgebras. The aim is to transform a coalgebra $\langle S, \alpha \rangle$ to a coalgebra $\langle S, \alpha' \rangle$ and define weak bisimulation on $\langle S, \alpha \rangle$ as strong bisimulation on $\langle S, \alpha' \rangle$. Our solution produces the same weak bisimulation notions as the concrete ones, when instantiated to some concrete systems (LTS and generative probabilistic systems).

Another weak branching-time semantic relation that has some attractive features is branching bisimulation [Gla93, GW96]. There are also several proposals for branching bisimulation for probabilistic systems in the literature [Seg95b, BH97, AW05]. However, we do not touch upon the topic of branching bisimulation in this thesis.

1.3 Comparing expressiveness of system types

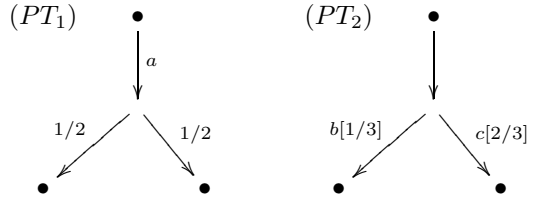
One of the main objectives in the thesis is to develop a method for formal comparison of different types of (probabilistic) systems. If we look at the fully deterministic labelled transition systems and the non-deterministic labelled transition systems, especially in their diagram representation, it is intuitively clear that the former are at most as expressive as the latter, i.e. the LTS can express any fully deterministic LTS and moreover they have the possibility of expressing non-determinism.

In particular, given a fully deterministic LTS $\langle S, \rightarrow \rangle$, for $\rightarrow$ a function from S to $A \times S$, we can transform it to an LTS $\langle S, \rightarrow' \subseteq S \times A \times S \rangle$ with the following definition of a transition relation $\rightarrow'$:

$$\langle s, a, t \rangle \in \rightarrow' \iff \rightarrow (s) = \langle a, t \rangle$$

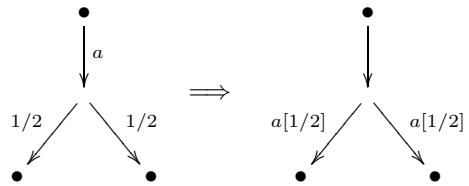
or equivalently, in terms of transition notation we have $s \xrightarrow{a}{}' t \iff s \xrightarrow{a} t$ which means that if we identify the systems with their diagrams, then the transformation is the identity. Still, the question is: does this make the LTS more expressive? What should be the formal notion of expressiveness?

Furthermore, consider the two types of probabilistic systems that allow for transitions as shown in the next diagram.



Hence, the systems with transitions of shape (PT_1) allow for a label followed by a probabilistic choice between the next states, where the systems with transitions of shape (PT_2) allow for a probabilistic choice that is distributed over both the action labels and the next states. The first type of systems are known as *simple Segala probabilistic automata* [SL94, Seg95b, Sto02a]; the second type of systems are the *general Segala probabilistic automata* with a generative behavior [GSST90, Seg95b].

It is intuitively clear that every transition of a simple Segala automaton of shape (PT_1), can be transformed to a transition of a general Segala automaton of shape (PT_2) by pushing the label into the probabilistic choice as shown in the next diagram.



Hence, we are intuitively convinced that indeed the general Segala systems are at least as general as the simple ones. However, the question of a formal proof of this fact remains open. In order to answer this question, we first of all need a criterion according to which we can compare the expressive power of two types of systems.

In accordance with the mentioned examples reflecting our intuition, we choose for the following expressiveness criterion: Let $\mathbf{C}_1$ and $\mathbf{C}_2$ denote two classes of systems. Each system of class $\mathbf{C}_1$ is defined as a pair of a set of states and a transition function, and the same holds for the systems in the class $\mathbf{C}_2$. We consider the class $\mathbf{C}_2$ at least as expressive as the class $\mathbf{C}_1$ if and only if there exists a translation map $\mathcal{T} : \mathbf{C}_1 \to \mathbf{C}_2$ such that:

- $\mathcal{T}$ leaves the state set unchanged;
- $\mathcal{T}$ is injective, i.e. different $\mathbf{C}_1$ systems are translated to different $\mathbf{C}_2$ systems;
- $\mathcal{T}$ preserves and reflects the corresponding notions of bisimilarity.

The last requirement imposed to the translation map corresponds to the requirement that the translations should be faithful copies of the originals that have the same behavior. To be more precise, assume $\sim_1$ and $\sim_2$ are the bisimilarity equivalences for the classes of systems $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively. Then we want that for any two states s and t, $s \sim_1 t$ in the original $\mathbf{C}_1$ system, if and only if $s \sim_2 t$ in the translated system in $\mathbf{C}_2$.

Considering the various types of systems as coalgebras, we show that categorial translations defined in terms of injective natural transformations satisfy the criteria imposed on the translation maps. Using this results we are able to build a hierarchy of the relevant probabilistic system types.

1.4 Results and overview

The main contributions of the thesis can be summarized as follows:

- We provide a comparative study of various existing types of probabilistic automata based models, their bisimulations, and their parallel composition operators (Chapter 2).
- We give a representation of many probabilistic system types as coalgebras of a functor, by specifying a set of inductively defined functors (Chapter 3 and Chapter 4).
- We prove, in a modular way, that for a broad class of functors coalgebraic bisimilarity coincides with the concrete one (Chapter 3 and Chapter 4).
- We prove that injective natural transformations lead to translations between coalgebras that preserve and reflect bisimilarity (Chapter 4).
- We build an expressiveness hierarchy of probabilistic models (informally presented in Chapter 2, full results and proofs in Chapter 4).

- We define a (parameterized) notion of weak bisimulation for action-type coalgebras (Chapter 5).
- We show that the concrete notions of weak bisimulation for LTS [Mil89] and for generative probabilistic systems [BH97] are covered by the coalgebraic definition (Chapter 5).
- We briefly consider two more semantic relations (simulation and colored transition equivalence) for coalgebras and probabilistic systems (Chapter 6).
- We study possible ways of composing coalgebras and defining paths in coalgebras (Chapter 6).

Summary

Chapter 2 presents a comparative overview of probabilistic systems that exist in the literature. It introduces the various types of probabilistic transition systems together with their concrete definitions of bisimulations. This chapter also presents a detailed study of parallel composition between the various probabilistic systems motivating the existence of some classes of probabilistic automata. The intuitive translations between the probabilistic models that lead to the expressiveness hierarchy are also discussed in this chapter.

Chapter 3 introduces the notions needed from category theory and coalgebras. Moreover, in this chapter we gather some small results of coalgebraic nature that are not exclusively connected to one of the following chapters. We introduce the class of functors that we work with throughout the thesis and we present a characterization of coalgebraic bisimilarity in terms of transfer conditions for our inductively defined functors. We note that, in contrast to what is common in the literature, when working with the probability distribution functor we do not impose the restriction of finite support. This is often required since there are proofs that the finitely supported distribution functor is well-behaved [VR99, Mos99], i.e. preserves weak pullbacks, and hence bisimilarity is an equivalence. We show that also the unrestricted probability distribution functor preserves weak pullbacks.

Chapter 4 presents the probabilistic systems as coalgebras of a functor from our class of functors and shows a direct correspondence between the concrete bisimulation definitions from Chapter 2 and the coalgebraic bisimulations induced by the functors. The main topic of this chapter is proving comparison results i.e. the hierarchy of the probabilistic system types. In order to establish the hierarchy we first prove the result that injective natural transformations between well-behaved functors lead to translations from one class of coalgebras to another that obey our comparison criterion. Then the hierarchy result follows by providing appropriate injective natural transformations. As far as we know, this form of application of the theory of coalgebras is not reported before in the literature.

Chapter 5 presents a study of a definition of weak bisimulation for coalgebras. We consider *action-type* coalgebras. These are coalgebras whose visible behavior includes performing actions. Any coalgebra can trivially be considered an action-type coalgebra. Since weak bisimulation abstracts from invisible behavior, we focus on abstracting away from invisible actions. If the coalgebras considered are not action-type or there are no invisible actions, then weak bisimulation amounts to strong bisimulation. Given an action-type system $\mathcal{S}$ with action set A, we adopt a two-phase approach in defining weak bisimulation: First, we define a "$*$-extension" of a given system which is a system that captures the behavior of $\mathcal{S}$ on finite words of actions. Next, we fix a set of invisible actions $\tau \subseteq A$ and transform the $*$-extension into a "weak-τ-extension" which is insensitive to τ steps. Then we define weak bisimilarity on $\mathcal{S}$ as strong bisimilarity on the weak-τ-extension. We justify this generalized approach by two important examples, the LTS and the generative probabilistic systems. A technical effort in this chapter is spent on showing that the coalgebraic and the concrete definition of weak bisimulation coincide for the case of generative probabilistic systems. This coincidence result also justifies the existing concrete definition of weak bisimulation for the generative probabilistic systems. Finally, it is worth mentioning that the concrete definition [BH97, BH99] restricts to finite systems, whereas our coalgebraic definition does not, and therefore our results extend the results of [BH97, BH99] to systems with an arbitrary number of states.

Chapter 6 is an open chapter presenting some questions and preliminary investigations for future work. We address several topics related to other semantic relations, such as: simulations, colored transition semantics, composition of coalgebras and ways to define paths in coalgebras. For simulations, there is a satisfactory coalgebraic and concrete treatment. Intrigued by [GW96] and [AW05], we look at the notions of consistent coloring and colored trace equivalence from a coalgebraic point of view, just to conclude that they coincide with the notion of homomorphism and behavioral equivalence. We also present a way of composing coalgebras, an operation that seems to be of use for obtaining trace semantics. Finally, we discuss possibilities for defining paths for coalgebras.

Origin of the chapters

Most of the material presented in this thesis was published before in several papers:

- Chapter 2 is a revised version of [SV04].
- Chapter 3 is mainly of introductory nature. Partial results on characterizing coalgebraic bisimilarity in terms of transfer conditions were previously published in [BSV03] and in [BSV04]. The result on weak pullback preservation of the unrestricted probability distribution functor was previously published in [SVW04]. The notions of action-type coalgebras and total weak pullbacks are also from [SVW05, SVW04].

- Chapter 4 is a revised version of the papers [BSV03] and [BSV04]. The revision mainly consists of providing full and modular proof of bisimulation correspondence that builds upon the material presented in Chapter 3.
- Chapter 5 is a revised version of [SVW05] and [SVW04], except for the parts of the mentioned papers that already found their way into Chapter 3.
- The last three sections of Chapter 6 with complete proofs were published as [Sok05].

1.5 Related work

Throughout the thesis we extensively refer to the literature and related work. In this section we briefly mention the work of other authors that has been most influential for this thesis.

Various probabilistic transition systems have been introduced and studied by many authors in many papers, among which are [Var85, PZ86, GSST90, Han91, Seg95b]. Bisimulation and bisimilarity for labelled transition systems [Par81, Mil89] were generalized to the probabilistic setting by Larsen and Skou [LS91]. Most of the concrete probabilistic transition systems come equipped with a notion of bisimulation that is based on the definition of Larsen and Skou. Parallel composition of various probabilistic systems was also studied by many authors. For a broad overview the reader is referred to [DHK98, Bai98]. Some comparison results for the expressive power of some probabilistic models are also reported in the literature: Van Glabbeek et al. compare the expressiveness power of the generative, the reactive and the stratified models [GSST90, GSS95]; the alternating model [Han91] and the non-alternating model i.e. the simple Segala probabilistic automata [Seg95b] are compared in [Sto02a, BS01, ST05].

The theory of universal coalgebras was systematically treated for the first time by Rutten [Rut96, Rut00]. Further introductory texts on the subject can be found in the articles by Jacobs and Rutten [JR96], Jacobs [Jac02] and Gumm [Gum99]. Probabilistic systems, in particular Markov chains, were represented as coalgebras by Rutten and de Vink [VR99], and by Moss [Mos99]. In these papers also correspondence of coalgebraic bisimulation [AM89] (for the distribution functor) and the Larsen and Skou bisimulation for Markov chains was independently established.

A notion of (parameterized) coalgebraic simulation was introduced by Jacobs and Hughes [JH03]. Various definitions of concrete (probabilistic) simulation also exist in the literature: for labelled transition systems by Milner [Mil89], for generative systems by Baier [Bai98], for simple Segala systems by Segala [Seg95b].

For labelled transition systems weak bisimulation is an established notion [Mil89]. For some types of probabilistic systems there also exist notions of weak bisimulation. Segala [SL94, Seg95b] proposed four notions of weak relations for his model of simple probabilistic automata. Baier and Hermanns [BH97, Bai98,

BH99] have given a rather appealing definition of weak bisimulation for generative probabilistic systems. Philippou, Lee and Sokolsky [PLS00] studied weak bisimulation in the setting of the alternating model [Han91]. This work was extended to infinite systems by Desharnais, Gupta, Jagadeesan and Panangaden [DGJP02b]. The same authors also provided a metric analogue of weak bisimulation [DGJP02a].

There has been early work on weak bisimulation for while programs in a coalgebraic setting by Rutten [Rut99], succeeded by a syntactic approach to weak bisimulation by Rothe [Rot02]. In the latter paper, weak bisimulation for a particular class of coalgebras was obtained by transforming a coalgebra into an LTS and making use of Milner's weak bisimulation there. This approach also enabled a definition of weak homomorphisms and weak simulation relations. Later, in the work of Rothe and Mašulović [RM02] a complex, but interesting coalgebraic theory was developed leading to weak bisimulation for functors that weakly preserve pullbacks. However, in the case of probabilistic and similar systems, it does not lead to intuitive results and can not be related to the concrete notions of weak bisimulation mentioned above. Our approach to weak bisimulation provides a (parameterized) notion of weak bisimulation for coalgebras that coincides with the concrete notions for labelled transition systems [Mil89] and for generative probabilistic systems [BH97, Bai98, BH99].

Our ways of composing coalgebras and defining paths are related to the work on trace-style semantics by Jacobs [Jac04] and Hasuo and Jacobs [HJ05b, HJ05a].

2

Probabilistic automata: system types, parallel composition and comparison

In this chapter we survey various notions of probabilistic automata and probabilistic bisimulation. The chapter provides an overview of existing models of probabilistic systems and explains the relationship between them. In addition, we discuss parallel composition for the presented types of systems.

The notion of a state machine has proved useful in many modelling situations, amongst others, the area of validation of probabilistic systems. In the literature up to now, many types of probabilistic automata have been proposed and many of these have been actually used for verification purposes. In this chapter we discuss a number of probabilistic automata with discrete probability distributions. For continuous-time probabilistic systems the interested reader is referred to [BDEP97, DEP98, D'A99, BHHK00, Hil94, Alf98]. Models of stochastic systems that are not represented by transition systems can also be found in [BLFG95] and [PA91]. Other models of stochastic concurrent systems are based on Petri nets (stochastic Petri nets, SPN, and generalized stochastic Petri nets, GSPN). The reader is referred to [Bal01] for an overview of Petri net based models.

Due to the variety of proposed models, it is often the case that results have to be interpreted from one type of systems to another. Therefore we compare the considered types of probabilistic automata in terms of their expressiveness. The comparison is achieved by placing a partial order on the classes of such automata, where one class is less than another if each automaton in the first class can be translated to an automaton of the other class such that translations both reflect and preserve the respective notions of bisimilarity. Hence, bisimulation and bisimilarity are central notions in this overview. Other comparison criteria are important as well, e.g. logical properties, logical characterization of bisimulation [LS91], complexity of algorithms for deciding bisimulation

[Bai98, BEMC99, CY95, Sto02a] and so on. We choose the comparison criterion formulated in terms of strong bisimulation because of its simplicity and because we work with labelled transition systems, for which bisimulation semantics arises naturally from the step-by-step behavior.

A major distinction of probabilistic automata is that between fully probabilistic vs. non-deterministic ones. In a fully probabilistic automaton every choice is governed by a probability distribution (over set of states or states combined with actions). The probability distribution captures the uncertainty about the next state. If we abstract away from the actions in a fully probabilistic automaton, we are left with a discrete time Markov chain. Subsequently, standard techniques can be applied to analyze the resulting Markov chains. Sometimes, the incomplete knowledge about the system behavior can not be represented probabilistically. We speak in this case of a non-deterministic probabilistic automaton. Most of the models that we consider include some form of non-determinism and hence fall in the category of non-deterministic probabilistic automata. As pointed out by various authors, e.g. [Hoa85, Seg95b, Alf97, Sto02b, AB02] non-determinism is essential for modelling scheduling freedom, implementation freedom, the external environment and incomplete information. Furthermore, non-determinism is essential for the definition of an asynchronous parallel composition operator that allows interleaving. Often two kinds of non-deterministic choices are mentioned in the literature (see e.g. [Sto02b, Har02]), *external* non-deterministic choices influenced by the environment, specified by having several transitions with different labels leaving from the same state, and *internal* non-determinism, exhibited by having several transitions with the same label leaving from a state. We use the term non-determinism for *full non-determinism* including both internal and external non-deterministic choices.

We introduce several classes of automata, ranging from the simplest models to more complex ones. The questions that we will address for each individual class are:

- the definition of the type of automaton and the respective notion of strong bisimulation;
- the relation of the model with other models;
- presence and form of non-determinism;
- the notion of a product or parallel composition in the model.

The set-up of the chapter is as follows: Section 2.1 presents the necessary notions considering probability theory, automata (transition systems), and concurrency theory, in particular compositional operators. In Section 2.2 we focus on the various definitions of probabilistic automata in isolation with their corresponding notions of bisimulation. In Section 2.3 the operators of parallel composition are discussed. We address the interrelationship between the introduced types of automata in Section 2.4. Some conclusions are briefly presented in Section 2.5.

2.1 Basic notions

2.1.1 Probability distributions

In this subsection we collect the basic definitions from probability theory that will be used throughout the thesis.

Definition 2.1.1. *Let Ω be a set. A function $\mu\colon \Omega \to \mathbb{R}_0^+$ is a* discrete probability distribution*, or distribution for short, on Ω if $\sum_{x\in\Omega} \mu(x) = 1$. The set $\{x \in \Omega |\ \mu(x) > 0\}$ is the* support *of μ and is denoted by* $\mathrm{supp}(\mu)$*. By $\mathcal{D}(\Omega)$ we denote the set of all discrete probability distributions on the set Ω.*

We note here that the sum of an arbitrary family $\{x_i \mid i \in I\}$ of non-negative real numbers is defined as $\sum_{i\in I} x_i = \sup\{ \sum_{i\in J} x_i \mid J \subseteq I, J \text{ finite } \}$. The following property will be used on several occasions throughout the thesis, and it justifies the name *discrete* probability distributions.

Proposition 2.1.2. *Let μ be a discrete probability distribution. Then the set* $\mathrm{supp}(\mu)$ *is at most countable.*

Proof Let $x \in \mathrm{supp}(\mu)$ for a distribution μ. Then $\mu(x) > 0$ and therefore there exists a natural number n such that $x > 1/n$. So we have, $\mathrm{supp}(\mu) \subseteq \cup_{n\in\mathbb{N}} \mathrm{supp}_n(\mu)$ where $\mathrm{supp}_n(\mu) = \{x \in \mathrm{supp}(\mu) \mid x > 1/n\}$. Now, since $\sum_{x\in\mathrm{supp}(\mu)} \mu(x) = 1$, the set $\mathrm{supp}_n(\mu)$ has less than n elements, for all $n \in \mathbb{N}$, i.e., it is finite. Therefore the set $\mathrm{supp}(\mu)$ is at most countable, being a countable union of finite sets. □

Hence the discrete probability distributions are indeed discrete, i.e. at most countably many elements of Ω are assigned non-zero probability. In the same way one obtains: if the sum of the values of a non-negative real valued function is finite, then the function has non-zero values at at most countably many elements of the domain.

For $\mu \in \mathcal{D}(\Omega)$ and $X \subseteq \Omega$ we denote $\mu[X] = \sum_{x\in X} \mu(x)$. If $x \in \Omega$, then μ_x^1 denotes the unique probability distribution with $\mu_x^1(x) = 1$, also known as the *Dirac distribution* for x. If μ is a distribution with finite support $\{s_1, \ldots, s_n\}$, we sometimes write $\{s_1 \mapsto \mu(s_1), \ldots, s_n \mapsto \mu(s_n)\}$. With this notation, $\mu_x^1 = \{x \mapsto 1\}$.

Definition 2.1.3. *Let $\mu_1 \in \mathcal{D}(S)$ and $\mu_2 \in \mathcal{D}(T)$. The product $\mu_1 \times \mu_2$ of μ_1 and μ_2 is a distribution on $S \times T$ defined by $(\mu_1 \times \mu_2)(s,t) = \mu_1(s) \cdot \mu_2(t)$, for $\langle s,t\rangle \in S \times T$.*

If $\mu \in \mathcal{D}(S \times T)$, we use the notation $\mu[s,T]$ for $\mu[\{s\} \times T]$ and $\mu[S,t]$ for $\mu[S \times \{t\}]$. We adopt from [JL91] the lifting of a relation between two sets to a relation between distributions on these sets.

Definition 2.1.4. *Let $R \subseteq S \times T$ be a relation between the sets S and T. Let $\mu \in \mathcal{D}(S)$ and $\mu' \in \mathcal{D}(T)$ be distributions. Define $\mu \equiv_R \mu'$ if and only if there exists a distribution $\nu \in \mathcal{D}(S \times T)$ such that*

1. *$\nu[s, T] = \mu(s)$ for any $s \in S$*
2. *$\nu[S, t] = \mu'(t)$ for any $t \in T$*
3. *$\nu(s, t) \neq 0$ implies $\langle s, t\rangle \in R$.*

The lifting of a relation R preserves the characteristic properties of preorders and equivalences (cf. [JLY01]). For the special case of an equivalence relation there is a simpler way to define the lifting (cf. [JLY01, Sto02b, Bai98]).

Proposition 2.1.5. *Let R be an equivalence relation on the set S and let $\mu, \mu' \in \mathcal{D}(S)$. Then $\mu \equiv_R \mu'$ if and only if $\mu[C] = \mu'[C]$ for all equivalence classes $C \in S/R$.* □

Lifting of a relation $R \subseteq S \times T$ to a relation $\equiv_{R,A} \subseteq \mathcal{D}(A \times S) \times \mathcal{D}(A \times T)$, for a fixed set A, will also be needed.

Definition 2.1.6. *Let R be a relation between S and T. Let $\hat{R} \subseteq (A \times S) \times (A \times T)$ be given by*

$$\langle\langle a, s\rangle, \langle a, t\rangle\rangle \in \hat{R} \iff \langle s, t\rangle \in R.$$

Then the lifted relation $\equiv_{R,A}$ is defined as $\equiv_{R,A} = \equiv_{\hat{R}}$.

From Proposition 2.1.5 we get the following corollary.

Corollary 2.1.7. *Let R be an equivalence relation on a set S, A a set, and let $\mu, \mu' \in \mathcal{D}(A \times S)$. Then*

$$\mu \equiv_{R,A} \mu' \iff \forall C \in S/R, \forall a \in A\colon \mu[a, C] = \mu'[a, C].$$

□

In later chapters we will use and treat liftings of relations in a general abstract setting.

2.1.2 Non-probabilistic automata, Markov chains, bisimilarity

The terms automaton, transition system or just system will be used as synonyms.

Non-probabilistic automata

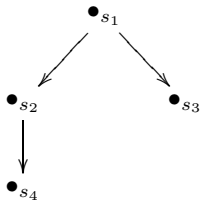
Definition 2.1.8. *A transition system, TS for short, is a pair $\langle S, \alpha \rangle$ where*

1. *S is a set of states*
2. *$\alpha : S \to \mathcal{P}(S)$ is a transition function.*

Here $\mathcal{P}(S)$ denotes the powerset of S. If $\langle S, \alpha \rangle$ is a transition system such that $s, s' \in S$ and $s' \in \alpha(s)$ we write $s \to s'$ and call it a transition.

Often in the literature a TS is given as a triple, including besides the set of states and the transition function also a subset of initial states, or a single initial state. Here we consider **no initial states** and therefore they are not present in the definition. Instead of a transition function one could equivalently consider a transition relation as a subset of $S \times S$. Our choice here is to always present the transitions via a **transition function**.

A common way of representing a TS is via its transition diagram. For example, the system $\langle S, \alpha \rangle$ where $S = \{s_1, s_2, s_3, s_4\}$ and $\alpha(s_1) = \{s_2, s_3\}$, $\alpha(s_2) = \{s_4\}$, $\alpha(s_3) = \alpha(s_4) = \emptyset$, is represented as follows:



The states s_3 and s_4 are *terminating* states, with no outgoing transitions.

It is often of use to model the phenomenon that a change of a state in a system happens as a result of executing an *action*. Therefore, labelled transition systems evolve from transition systems. There are two ways to incorporate labels in a TS: by labelling the states (usually with some values of variables, or a set of propositions that hold in a state), or by explicitly labelling the transitions with actions or action names. We focus on systems with **labels on the transitions**.

Definition 2.1.9. *A labelled transition system (LTS), or a non-deterministic automaton, is a triple $\langle S, A, \alpha \rangle$ where*

1. *S is a set of states*
2. *A is a set of actions*
3. *$\alpha : S \to \mathcal{P}(A \times S)$ is a transition function.*

If $\langle S, A, \alpha \rangle$ is an LTS, then the transition function α can equivalently be considered as a function from S to $\mathcal{P}(S)^A$, the collection of functions from A to $\mathcal{P}(S)$.

As in the case of TSs, for any state $s \in S$ of an LTS, every element $\langle a, s' \rangle \in \alpha(s)$ determines a transition which is denoted by $s \xrightarrow{a} s'$.

The class of LTSs (non-deterministic automata) is denoted by **LTS**. Deterministic automata, given by the next definition, form a subclass of **LTS**.

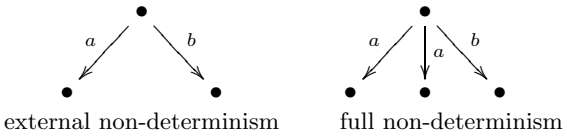
Definition 2.1.10. *A deterministic automaton is a triple $\langle S, A, \alpha \rangle$ where*

1. *S is a set of states*
2. *A is a set of actions*
3. *$\alpha : S \to (S + 1)^A$ is a transition function.*

Notation 2.1.11. We denote by $+$ the disjoint union of two sets. More precisely, $A + B = \{\langle a, 1 \rangle \mid a \in A\} \cup \{\langle b, 2 \rangle \mid b \in B\}$, but we simply consider that $A + B$ is the union of two disjoint copies of the sets A and B. The set 1 is a singleton set containing the special element $*$, i.e. $1 = \{*\}$. We assume that $* \notin S$. The notation $(S + 1)^A$ stands for the collection of all functions from A to $S + 1$.

The special set 1 and the disjoint union construction allow for writing partial functions as functions. In a deterministic automaton each state s is assigned a function $\alpha(s) \colon A \to S + 1$, which can also be considered a partial function from the set of actions to the set of states, meaning that whenever $\alpha(s)(a) = s'$ for some $s' \in S$, hence $\alpha(s) \neq *$, then the transition $s \xrightarrow{a} s'$ is enabled in s. We denote the class of all deterministic automata by **DLTS**.

We note that the class of automata **DLTS** exhibits external non-determinism, while in **LTS** there is full non-determinism. Namely, in **DLTS** (see the left diagram below) multiple transitions are possible in a state only if they have different labels. Hence, the nondeterministic choice is made only between the labels offered in a state and this choice is typically a choice of the environment. On the other hand, in **LTS** (see the right diagram below) it is also possible to have multiple outgoing transitions from a state labelled with the same label. This characterizes full non-determinism. Beside the non-deterministic choice of the environment on which label is offered, the automaton itself has a non-deterministic choice of deciding a next state after a transition with a given label.



external non-determinism full non-determinism

In the introduction (Section 1.1) we also mentioned fully deterministic transition systems in which only a single transition is possible in each state. These systems do not allow for any type of non-determinism.

Markov chains

The simplest class of fully probabilistic automata is the class of discrete time Markov chains. The theory of Markov chains is rich and huge (see, e.g., [KS76, How71, BH01, Hav01]) and we only provide a simple definition of a discrete time Markov chain here.

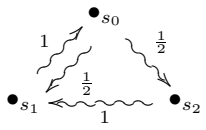
Definition 2.1.12. *A Markov chain is a pair* $\langle S, \alpha \rangle$ *where*

1. S *is a set of states*
2. $\alpha : S \to \mathcal{D}(S)$ *is a transition function.*

where $\mathcal{D}(S)$ *is defined in Definition 2.1.1.*

Markov chains evolve from transition systems, when probability is added to each transition such that for any state the sum of the probabilities of all outgoing transitions equals 1. The class of all Markov chains is denoted by **MC**. If $s \in S$ and $\alpha(s) = \mu$ with $\mu(s') = p > 0$ then the Markov chain $\langle S, \alpha \rangle$ is said to go from a state s with probability p to a state s'. Notation: $s \rightsquigarrow \mu$ and $s \overset{p}{\rightsquigarrow} s'$.

Example 2.1.13. The following diagram represents an example Markov chain $\langle S, \alpha \rangle$.



$$S = \{s_0, s_1, s_2\}$$
$$\alpha(s_0) = \{s_0 \mapsto 0, s_1 \mapsto \tfrac{1}{2}, s_2 \mapsto \tfrac{1}{2}\}$$
$$\alpha(s_1) = \mu^1_{s_0}$$
$$\alpha(s_2) = \mu^1_{s_1}$$

Bisimulation and bisimilarity

Different semantics or notions of behavior can be given to labelled transition systems. We work with bisimulation semantics (Park [Par81], Milner [Mil83, Mil89]): two states in a system represented by an LTS are equivalent whenever there exists a bisimulation relation that relates them. A bisimulation relation compares the one-step behavior of two states and has a nice extension to the probabilistic case (as explored in [LS91]). In [JS90] probabilistic extensions of a number of other well known process equivalences have been studied like probability trace, completed trace, failure and ready equivalence. Other probabilistic process equivalences are probabilistic simulation and bisimulation introduced by Segala and Lynch [SL94, Seg95b], Yi and Larsen's testing equivalence [YL92], and CSP equivalences of Morgan et al. [MMSS96], Lowe [Low95] and Seidel [Sei95]. An overview of several probabilistic process equivalences can be found in [LN04].

Definition 2.1.14. *Let* A *be a set of actions and* $\langle S, A, \alpha \rangle$ *and* $\langle T, A, \beta \rangle$ *be two LTSs. A relation* $R \subseteq S \times T$ *is a* bisimulation *relation if* $\langle s, t \rangle \in R$*, implies for*

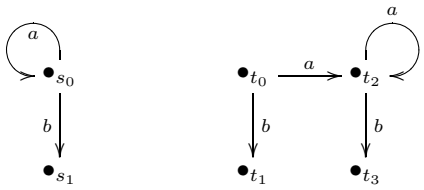
all $a \in A$ that

if $s \xrightarrow{a} s'$, then there exists $t' \in T$ such that $t \xrightarrow{a} t'$ and $\langle s', t' \rangle \in R$, and

if $t \xrightarrow{a} t'$, then there exists $s' \in S$ such that $s \xrightarrow{a} s'$ and $\langle s', t' \rangle \in R$.

Let $s \in S$ and $t \in T$. The states s and t are called bisimilar*, denoted by $s \sim t$ if there exists a bisimulation relation R with $\langle s, t \rangle \in R$.*

Example 2.1.15. For the following LTSs we have, for example, $s_0 \sim t_0$ since $R = \{\langle s_0, t_0 \rangle, \langle s_0, t_2 \rangle, \langle s_1, t_1 \rangle, \langle s_1, t_3 \rangle\}$ is a bisimulation.



Remark 2.1.16. Instead of comparing states in two systems $\langle S, A, \alpha \rangle$ and $\langle T, A, \beta \rangle$ we can always consider one joined system $\langle S+T, A, \gamma \rangle$ with $\gamma(s) = \alpha(s)$ for $s \in S$ and $\gamma(t) = \beta(t)$ for $t \in T$. We also write $\gamma = \alpha + \beta$. Therefore, bisimulation can be defined as a relation on the set of states of a system. Furthermore, if $R \subseteq S \times S$ is a bisimulation, then the equivalence closure of R is also a bisimulation. Hence, bisimilarity $\sim$ is not affected by the choice of defining bisimulation as an equivalence.

Definition 2.1.17. *An equivalence relation R on a set of states S of an LTS is an equivalence bisimulation if it is an equivalence and a bisimulation. The states s and t are called* e-bisimilar*, denoted by $s \sim_e t$, if there exists an equivalence bisimulation R with $\langle s, t \rangle \in R$.*

By Remark 2.1.16, the following proposition holds.

Proposition 2.1.18. *Let $\langle S, A, \alpha \rangle$ and $\langle T, A, \beta \rangle$ be two LTSs, and let $s \in S$, $t \in T$. Then $s \sim t$ if and only if $s \sim_e t$ in $\langle S + T, A, \alpha + \beta \rangle$.*

□

Bisimulation on **DLTS** is defined exactly the same as for **LTS** i.e. with Definition 2.1.17.

The standard notion of probabilistic bisimulation is the one introduced by Larsen and Skou [LS91] originally formulated for reactive systems (see Section 2.2.1). An early reference to probabilistic bisimulation can be found in [BM89]. In the case of Markov chains, bisimulation corresponds to ordinary lumpability of Markov chains [KS76, Buc94, Her98].

The idea behind probabilistic bisimulation is as follows. Since bisimilar states are considered "the same", it does not matter which element within a bisimulation class is reached. Hence, a bisimulation relation should compare the probability to reach an equivalence class and not the probability to reach a single state. In order to define bisimulation for Markov chains the lifting of a relation on a state S to a relation on $\mathcal{D}(S)$, as defined in Definition 2.1.4 and explained with Proposition 2.1.5, is used. Note that the comments of Remark 2.1.16 are in place here as well.

Definition 2.1.19. *An equivalence relation R on a set of states S of a Markov chain $\langle S, \alpha \rangle$ is a bisimulation if and only if $\langle s, t \rangle \in R$ implies*

$$\textit{if } s \rightsquigarrow \mu, \textit{ then there is a distribution } \mu' \textit{ with } t \rightsquigarrow \mu' \textit{ and } \mu \equiv_R \mu'.$$

The states s and t are called bisimilar*, denoted by $s \sim t$, if there exists a bisimulation R with $\langle s, t \rangle \in R$.*

Definition 2.1.19 will be used, with some variations, for defining bisimulation relations for all types of probabilistic automata that we consider in this thesis. However, note that in the case of Markov chains any two states of any two Markov chains are bisimilar, according to the given definition, since $\nabla = S \times S$ is a bisimulation on the state set of any Markov chain $\langle S, \alpha \rangle$. Namely, let $\langle S, \alpha \rangle$ be a Markov chain and $s, t \in S$, such that $\alpha(s) = \mu, \alpha(t) = \mu'$, i.e., $s \rightsquigarrow \mu, t \rightsquigarrow \mu'$. Then for the only equivalence class of ∇, S, we have $\mu[S] = 1 = \mu'[S]$ i.e. $\mu \equiv_R \mu'$ which makes $s \sim t$. This phenomenon can be explained with the fact that bisimilarity compares the observable behavior of two states in a system and the Markov chains are very simple systems in which there is not much to observe. Therefore there is an occasion to enrich Markov chains with actions or at least termination.

Notational matters

In Section 2.2 we will introduce ten other types of probabilistic automata, with corresponding notions of bisimulation. In order to avoid repetition of definitions we collect the following.

- A type of automata will always be a triple $\langle S, A, \alpha \rangle$ where S is a set of states, A is a set of actions and α is a transition function. The difference between the system types is expressed with the difference in the codomains of the corresponding transition functions.

- A bisimulation relation will always be defined as an equivalence on the set of states of a system. Depending on the type of systems the "transfer conditions" in the definition of bisimulation vary.

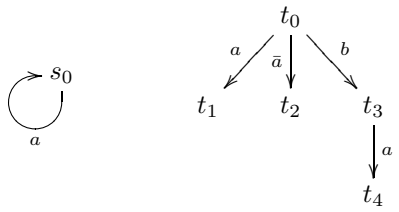
- For a particular type of system, the bisimilarity relation, denoted by $\sim$ is defined by: $s \sim t$ if and only if there exists a bisimulation R that relates s and t, i.e. $\langle s, t\rangle \in R$. Although we use the same notation $\sim$ for bisimilarity in different types of systems, it should be clear that for each type of systems, $\sim$ is a different relation.

2.1.3 Parallel composition of LTSs and MCs

Compositional operators serve the need of modular specification and verification of systems. They arise from process calculi, such as CCS ([Mil89]), CSP ([Hoa85]) and ACP ([BK85]), where process terms are built from atomic process terms with the use of compositional operators. Usually a model of a process calculus is a suitable class of transition systems. Therefore it is often the case that process terms are identified with their corresponding transition systems, and the compositional operators of the process calculus can be considered as operators for combining transition systems. Here we focus on the parallel composition operator. The definition of parallel composition varies a lot throughout different process calculi. In this section we consider the non-probabilistic case (LTSs) in order to explain variants of different parallel compositions, and the parallel composition of Markov chains in order to present the basics of probabilistic parallel composition.

Labelled transition systems

A major distinction between different parallel composition operators is whether they are *synchronous*, where the components are forced to synchronize whenever they can, or *asynchronous* where the components can either synchronize or act independently. Furthermore, different approaches for synchronization exist. The result of the parallel composition of two automata $\mathcal{A}_1 = \langle S_1, A, \alpha_1\rangle$ and $\mathcal{A}_2 = \langle S_2, A, \alpha_2\rangle$ is an automaton $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S_1 \times S_2, A, \alpha\rangle$ where the definition of α varies. Instead of a pair $\langle s, t\rangle \in S_1 \times S_2$ we will write $s\|t$ for a state in the composed automaton. Throughout this subsection we will use as running example, the parallel composition of the following two automata.

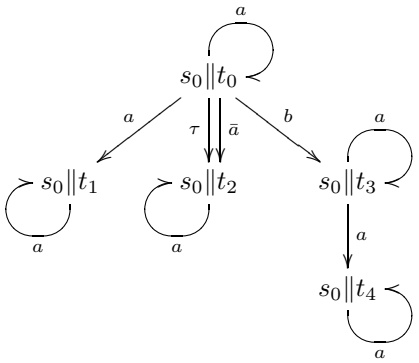


CCS style: The set of actions in this case contains compatible actions $a, \bar{a} \in A$ and a special idle or internal action $\tau \in A$. If one of the automata in state s can perform an action a changing to a state s' and the other one in state t can perform a's compatible action $\bar{a}$ moving to state t' then the composite automaton in state $s\|t$ can perform the idle action τ and move to state $s'\|t'$.

Furthermore, independent behavior of each of the automata is possible within the composed automaton.

$s\|t \xrightarrow{a} s'\|t'$ if and only if

1. $s \xrightarrow{b} s', t \xrightarrow{\bar{b}} t', a = \tau$, for b and $\bar{b}$ compatible actions, or
2. $s \xrightarrow{a} s'$ and $t' = t$, or
3. $t \xrightarrow{a} t'$ and $s' = s$.

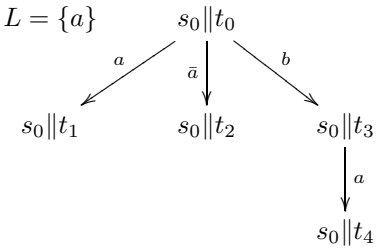


The presented CCS parallel composition is asynchronous. A synchronous variant (SCCS [Mil83]) is defined by omitting clauses 2. and 3. in the definition above (with a modified, total notion of compatibility).

CSP style: Communication or synchronization in a CSP style parallel composition occurs on a set of synchronizing actions. Thus actions that are intended to synchronize are listed in a set $L \subseteq A$ and the rest of the actions can be performed independently.

$s\|t \xrightarrow{a} s'\|t'$ if and only if

1. $s \xrightarrow{a} s'$ and $t \xrightarrow{a} t'$ and $a \in L$, or
2. $s \xrightarrow{a} s'$, $t = t'$ and $a \notin L$, or
3. $t \xrightarrow{a} t'$, $s = s'$ and $a \notin L$.



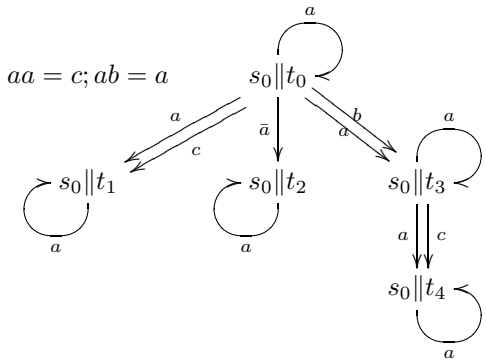
This type of parallel composition operator is synchronous for $L = A$, expresses only interleaving (shuffling) composition if $L = \emptyset$ and is never fully asynchronous with both independent behavior and communication allowed. An asynchronous CSP style parallel composition can be defined by omitting the clause "$a \notin L$" in clauses 2. and 3. above. In case of different action sets A_1 and A_2, of the two component automata, L is taken to be a subset of $A_1 \cap A_2$. If $L = A_1 \cap A_2$ then we say that synchronization on common actions occurs.

ACP style: In ACP, parallel composition is fully asynchronous, allowing both interleaving (independent behavior) and synchronization via a communication function. A communication function is a commutative and associative partial function $\gamma : A \times A \hookrightarrow A$. Instead of $\gamma(a, b)$ we will write ab.

For our running example in ACP style, let the communication function γ be the smallest commutative and associative partial function such that $\gamma(a, a) = aa = c$ and $\gamma(a, b) = ab = c$. It is easy to see that such a partial function exists.

$s\|t \xrightarrow{a} s'\|t'$ if and only if

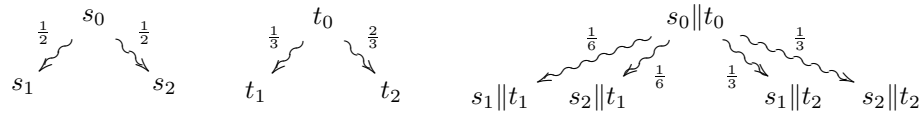
1. $s \xrightarrow{b} s'$ and $t \xrightarrow{c} t'$ with $bc = a$ defined, or
2. $s \xrightarrow{a} s'$, $t = t'$, or
3. $t \xrightarrow{a} t'$, $s = s'$.



Note that if A contains compatible actions and an idle action τ, and if $a\bar{a} = \tau$ for any compatible $a, \bar{a} \in A$ and undefined otherwise, then the ACP parallel composition operator specializes to the CCS parallel composition operator. On the other hand, for $aa = a$, $(a \in L \subseteq A)$ we get the asynchronous variant of the CSP parallel composition operator. If clauses 2. and 3. are dropped from the definition, we get a synchronous variant of the ACP parallel composition operator called communication merge.

Markov chains

Let $\mathcal{M}_1 = \langle S_1, \alpha_1 \rangle, \mathcal{M}_2 = \langle S_2, \alpha_2 \rangle$ be two Markov chains. Their parallel product is the Markov chain $\mathcal{M}_1 \| \mathcal{M}_2 = \langle S_1 \times S_2, \alpha \rangle$, where $\alpha(s\|t) = \alpha_1(s) \times \alpha_2(t)$, $\times$ denoting the product of distributions. Hence $s\|t \rightsquigarrow \mu$ if and only if $s \rightsquigarrow \mu_1, t \rightsquigarrow \mu_2$ and $\mu = \mu_1 \times \mu_2$. A small example of parallel composition of Markov chains is given in the next diagram.



Note that the parallel composition of two Markov chains is synchronous, since each step in the composed automaton consists of independent steps performed by each of the components. The way of defining the product of two distributions goes in favor of the interpretation that when put in parallel, each of the automata independently chooses its transition that contributes to a transition in the composed automaton.

2.2 Probabilistic models

This section defines the advanced types of probabilistic automata. The automata types are grouped in several subsections reflecting their common properties. Basically, every type of probabilistic automata arises from the plain

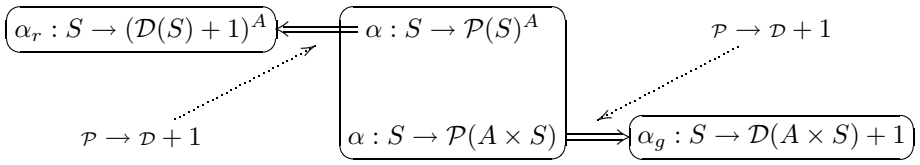
definition of a transition system with or without labels. Probabilities can then be added either to every transition, or to transitions labelled with the same action, or there can be a distinction between probabilistic and ordinary (non-deterministic) states, where only the former ones include probabilistic information, or the transition function can be equipped with structure that provides both non-determinism and probability distributions.

Each kind of probabilistic automata comes equipped with a notion of bisimulation, and all these notions, frequently only subtly different, will also find their way in this section.

2.2.1 Reactive, generative and I/O probabilistic automata

Two classical extensions of LTSs with probabilities are the reactive and the generative model. Throughout the years a large amount of research has been devoted to reactive and generative probabilistic systems. It is hard to note who introduced these systems first, but the reactive model was treated e.g. in [LS91, LS92, GSST90, GSS95], the generative in e.g. [GSST90, GSS95, Har02, HV02, CSZ92, Chr90, CC91], and the classification of these systems together with a so-called stratified model was proposed in [GSS95, GSST90].

The way these models arise from LTSs, by changing the transition function, can be explained with the following figure, where α denotes the transition function of an LTS, α_r and α_g the transition function of a reactive and a generative system, respectively.



The figure points out that the LTS can be defined by two types of transition functions (shown in the central box). In both of these types of transition functions the powerset construct $\mathcal{P}$ is used. If we change the powerset construct to the distribution construct with the termination possibility, $\mathcal{D}+1$, the one type changes to reactive systems (shown in the left box) and the other type changes to the type of generative systems (shown in the right box).

Definition 2.2.1. *A reactive probabilistic automaton is a triple $\langle S, A, \alpha \rangle$ where the transition function is given by*

$$\alpha : S \to (\mathcal{D}(S)+1)^A.$$

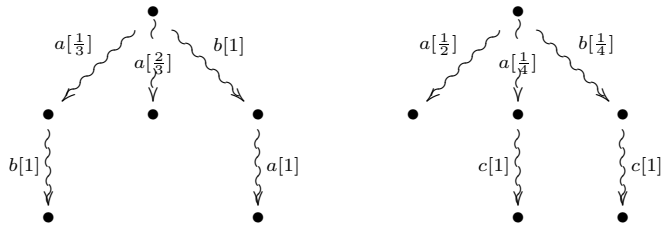
If $s \in S$ and $\alpha(s)(a) = \mu$ then we write $s \xrightarrow{a}\!\!\rightsquigarrow \mu$. More specifically, if $s' \in \mathrm{supp}(\mu)$, $\mu(s') = p$ we write $s \overset{a[p]}{\rightsquigarrow} s'$.

A generative probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ with a transition function

$$\alpha : S \to \mathcal{D}(A \times S) + 1.$$

When $s \in S$ and $\alpha(s) = \mu \in \mathcal{D}(A \times S)$ then we write $s \rightsquigarrow \mu$. More particularly, if $\langle a, s'\rangle \in \mathrm{supp}(\mu)$ with $\mu(\langle a, s'\rangle) = p$ we write $s \overset{a[p]}{\rightsquigarrow} s'$. We use $s \not\rightsquigarrow$ to denote that $\alpha(s) = *$.

Remark 2.2.2. In Definition 2.2.1 both uses of the special singleton set 1 appear. The first one, as in Definition 2.1.10 helps expressing partial functions. The second one, in the definition of a generative transition function, expresses the possibility of termination. If s is a state in a generative system with $\alpha(s) = *$ then s is a terminating state allowing no transition. For LTSs, termination is allowed by the fact that $\emptyset \in \mathcal{P}(A \times S)$. Hence, when changing from subsets to distributions, $*$ is added to play the role of the $\emptyset$.



Reactive system Generative system

In a reactive system probabilities are distributed over the outgoing transitions labelled with the *same action*, while in a generative system probabilities are distributed over *all* outgoing transitions from a state. A motivation for making this distinction is the different treatment of actions. In a reactive system actions are treated as *input* actions being provided by the environment. When a reactive system receives input from the environment, then it acts probabilistically by choosing the next state according to a probability distribution assigned to this input. There are no probabilistic assumptions about the behavior of the environment. On the other hand, in a generative system, as the name suggests, actions are treated as *output* generated by the system. When a generative system is in a state s it chooses the next transition according to the probability distribution $\alpha(s)$ assigned to s. The transition being chosen, the system moves to another state while generating the output action which labels this transition. Note that in a generative system there is no non-determinism present, while in a reactive system there is only external non-determinism, as in **DLTS**. We denote by **React** and **Gen** the classes of reactive and generative probabilistic automata, respectively.

Definition 2.2.3. *An equivalence relation R on S is a bisimulation on the reactive probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if $\langle s, t\rangle \in R$ implies, for all actions $a \in A$, that*

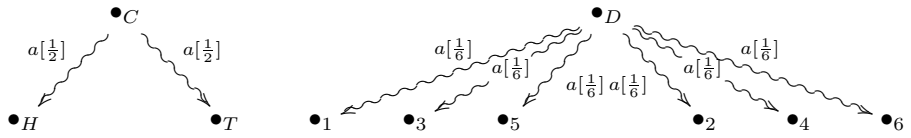
if $s \xrightarrow{a} \rightsquigarrow \mu$, then there exists a distribution μ' with $t \xrightarrow{a} \rightsquigarrow \mu'$ and $\mu \equiv_R \mu'$.

Above we have used the lifting from Definition 2.1.4. In order to state the definition of bisimulation for generative systems, the lifting from Definition 2.1.6 is used.

Definition 2.2.4. *An equivalence relation R on S is a bisimulation on the generative probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if for all $\langle s,t\rangle \in R$:*

if $s \rightsquigarrow \mu$, then there exists a distribution μ' with $t \rightsquigarrow \mu'$ and $\mu \equiv_{R,A} \mu'$.

Example 2.2.5. The equivalence relation R generated by the pairs $\langle C, D\rangle$, $\langle H, 1\rangle$, $\langle H, 3\rangle$, $\langle H, 5\rangle$, $\langle T, 2\rangle$, $\langle T, 4\rangle$, $\langle T, 6\rangle$ is a bisimulation for the probabilistic automaton given below. Hence, $C \sim D$. Note that this particular automaton belongs to both **React** and **Gen**.



An intuitive interpretation of this example is obtained by adding meaning "flip" to the action a in the left sub-automaton and a meaning "roll" to the action a in the right sub-automaton. Then the state C represents flipping of a fair coin, and the state D represents rolling a fair dice. The bisimilarity of the states C and D shows that it is the same whether one flips a fair coin or rolls a fair dice being interested only in whether, e.g., the outcome is odd or even.

I/O probabilistic automata

The model of input/output probabilistic automata, introduced by Wu, Smolka and Stark in [WSS97], exploiting the input/output automata by Lynch and Tuttle (cf. [LT87]), presents a combination of the reactive and the generative model.

Definition 2.2.6. *An input/output probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where*

1. *the set of actions A is divided into input and output actions, $A = A^{in} + A^{out}$*
2. *$\alpha : S \to \mathcal{D}(S)^{A^{in}} \times (\mathcal{D}(A^{out} \times S) + 1) \times \mathbb{R}_0^+$ is the transition function.*

The third component in the transition function assigns an output delay rate *to each state. If $s \in S$, then $\alpha(s) = \langle f^{in}, \mu^{out}, \delta_s\rangle$. We have that $\delta_s = 0$ if and only if $\mu^{out} = *$ i.e. delay is assigned to the states that generate output.*

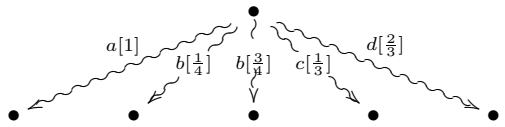
In an I/O automaton for every input action there is a reactive transition. Note that f^{in} is always a function and not a partial function as in the reactive model.

Hence each input action is enabled in each state of an I/O probabilistic automaton. The output actions are treated generatively. At most one generative probabilistic transition gives the output behavior of each state. The delay rate parameter δ_s is an aspect from continuous-time systems, and its meaning will become clear in Section 2.3 when we discuss compositions of I/O automata.

Denote the class of I/O automata by **IO**. We use a similar notation for transitions as in the reactive and the generative model. If $s \in S$ with $\alpha(s) = \langle f^{in}, \mu^{out}, \delta_s \rangle$ then

- if $a \in A^{in}$ with $f^{in}(a) = \mu$ we write $s \xrightarrow{a} \!\!\rightsquigarrow \mu$, furthermore, if $s' \in \text{supp}(\mu)$ with $\mu(s') = p$ we write $s \overset{a[p]}{\rightsquigarrow} s'$.

- if $\mu^{out} \neq *$ we write $s \rightsquigarrow \mu^{out}$ and if $\mu^{out}(a, s') = p > 0$ we write $s \overset{a[p]}{\rightsquigarrow} s'$.



transitions from a state in an I/O probabilistic automaton
$A^{in} = \{a, b\},\ A^{out} = \{c, d\}$

The I/O automata will not be compared and placed in the hierarchy of Section 2.4 since they involve a continuous element. It is obvious that, when ignoring the 0 delays, for $A^{out} = \emptyset$ one gets the reactive model (with all actions enabled) and for $A^{in} = \emptyset$ one gets the generative model with a delay rate assigned to each state. A connection exists between I/O automata and some models with structured transition relation (Section 2.2.3). Combined systems similar to I/O automata appear as models of process terms in the process algebra EMPA [Ber99, BG98]. In a recent work by Cheung and Hendriks a brand new model of probabilistic systems with I/O distinction was proposed [CH05].

Since we do not compare I/O automata in Section 2.4, we do not need a notion of bisimulation for them, although it can be defined by combining the transfer conditions for reactive and generative bisimulation, and taking care of the delay rate. In [WSS97] no notion of bisimulation is introduced. A different, rather complex, notion of behavior of I/O automata is considered which is beyond the scope of this thesis. A definition of bisimulation for I/O automata can be found in [SCS03].

2.2.2 Automata with different types of states

So far, we have seen some types of automata that allow modelling of probabilistic behavior, but none of those has the capability of also modelling full

non-determinism. The types of systems introduced in a minute allow full non-determinism while making a distinction between probabilistic states with outgoing probabilistic transitions, and non-deterministic states with action labelled transitions.

Stratified probabilistic automata

The simplest system with a distinction on states appears under the name of stratified probabilistic automaton, and is discussed in [GSS95, GSST90, SS90, HV98]. Stratified automata do not yet allow any form of non-determinism although there is a distinction on states.

Definition 2.2.7. *A stratified probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where the transition function α is given by*

$$\alpha : S \to \mathcal{D}(S) + (A \times S) + 1$$

The class of all stratified automata we denote by **Str**. Due to the disjoint union in the codomain of the transition function, there are three types of states in a stratified automaton: probabilistic states consisting of $s \in S$ such that $\alpha(s) \in \mathcal{D}(S)$, deterministic states $s \in S$ for which $\alpha(s) = \langle a, s'\rangle$ allowing a single action labelled transition, and terminating states $s \in S$ with $\alpha(s) = *$.

Definition 2.2.8. *An equivalence relation R on S is a bisimulation on the stratified probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if $\langle s, t\rangle \in R$ implies that*

1. *if $s \rightsquigarrow \mu$, then there exists a distribution μ' with $t \rightsquigarrow \mu'$ and $\mu \equiv_R \mu'$*
2. *if $s \xrightarrow{a} s'$, then there exists t' such that $t \xrightarrow{a} t'$ and $\langle s', t'\rangle \in R$.*

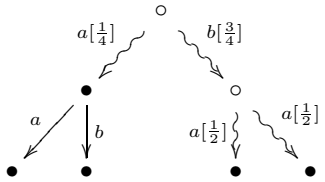
As a consequence of the definition, if $s \sim t$ in a stratified automaton and if $\alpha(s) = *$, then also $\alpha(t) = *$.

Vardi probabilistic automata

One of the earliest models of probabilistic automata was introduced by Vardi in [Var85] under the name *concurrent Markov chains*. The original definition of a concurrent Markov chain was given in terms of state labelled transition systems, for purposes of verification of logical properties. Therefore we slightly modify the definition, calling this class of automata Vardi probabilistic automata.

Definition 2.2.9. *A Vardi probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where the transition function α is given by*

$$\alpha : S \to \mathcal{D}(A \times S) \cup \mathcal{P}(A \times S)$$



Vardi probabilistic automaton

Remark 2.2.10. Note that $\cup$ is used in Definition 2.2.9 rather than $+$. One could consider the union disjoint, but it is of more use to identify $\mu^1_{\langle a,s'\rangle}$ with the singleton $\{\langle a, s'\rangle\}$, i.e. a state with a transition $s \overset{a[1]}{\rightsquigarrow} s'$ can be identified with a state allowing only one transition $s \xrightarrow{a} s'$.

In Vardi automata, the probabilistic states are of a generative kind, while the other states are non-deterministic with full non-determinism, as in an LTS. Therefore, the definition of bisimulation is a combination of Definition 2.1.17 and Definition 2.2.4.

Definition 2.2.11. *An equivalence relation R on S is a bisimulation on the Vardi probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if $\langle s, t\rangle \in R$ implies that*

1. *if $s \rightsquigarrow \mu$, then there exists a distribution μ' with $t \rightsquigarrow \mu'$ and $\mu \equiv_{R,A} \mu'$*
2. *if $s \xrightarrow{a} s'$, then there exists t' such that $t \xrightarrow{a} t'$ and $\langle s', t'\rangle \in R$.*

Remark 2.2.12. We note that in the literature, in particular in [Var85], there is no definition of bisimulation. However, the current understanding of probabilistic bisimulation, and the concept of a general coalgebraic definition of bisimulation allow us to state the previous definition.

We denote the class of Vardi probabilistic automata by **Var**.

The alternating models of Hansson

Another model that treats separately (purely) probabilistic and non-deterministic states is the alternating model introduced by Hansson, see for example [Han91, HJ94]. We present the class of alternating probabilistic automata **Alt**, its subclass of strictly alternating probabilistic automata **SA** and, in turn, two subclasses of **SA**, denoted by $\mathbf{SA}_n$ and $\mathbf{SA}_p$.

Definition 2.2.13. *An alternating probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where*

$$\alpha : S \to \mathcal{D}(S) + \mathcal{P}(A \times S).$$

The class of alternating automata is denoted by **Alt**. *Denote by N and P the subsets of S containing non-deterministic and probabilistic states, respectively.*

A strictly alternating automaton is an alternating automaton where for all $s \in S$ the following holds:

1. *if $s \in P$ with $\alpha(s) = \mu \in \mathcal{D}(S)$ then $\mathrm{supp}(\mu) \subseteq N$;*
2. *if $s \in N$ then for all $\langle a, s' \rangle \in \alpha(s)$, $s' \in P$.*

The class of all strictly alternating automata is denoted by **SA**.

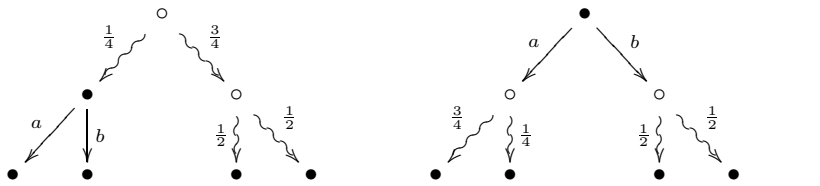
An automaton of **SA** *belongs to* $\mathbf{SA}_n$ *if and only if*

$$\forall s \in S \colon (\forall s' \in S, \forall a \in A, \forall p \in [0,1] \colon s' \overset{p}{\not\rightarrow} s \wedge s' \overset{a}{\not\rightarrow} s) \Rightarrow s \in N. \tag{2.1}$$

An automaton of **SA** *belongs to* $\mathbf{SA}_p$ *if and only if*

$$\forall s \in S \colon (\forall s' \in S, \forall a \in A, \forall p \in [0,1] \colon s' \overset{p}{\not\rightarrow} s \wedge s' \overset{a}{\not\rightarrow} s) \Rightarrow s \in P. \tag{2.2}$$

The classes **SA** [Han91, HJ94] and $\mathbf{SA}_n$ [And99, And02] are well-known, but we have chosen to present all these classes structurally. The class **Alt** is a slight generalization of the class **SA** and is similar to the stratified and Vardi models. In an alternating automaton only a distinction on states is imposed. In the strictly alternating model it is required that all successors of a non-deterministic state are probabilistic states and vice versa. Furthermore, the two subclasses $\mathbf{SA}_n$ and $\mathbf{SA}_p$ take care that any "initial state" is non-deterministic (2.1) and probabilistic (2.2), respectively. We define the subclasses $\mathbf{SA}_n$ and $\mathbf{SA}_p$ in order to make a precise comparison of the class **SA** with some of the other models (see Section 2.4).



alternating probabilistic automaton strictly alternating automaton ($\mathbf{SA}_n$)

For all the introduced classes of alternating automata, a single definition of bisimulation can be given, where the transfer conditions are exactly the same as for the stratified model, given in Definition 2.2.8.

2.2.3 Probabilistic automata with structured transitions

In this subsection we focus on three types of probabilistic automata that provide orthogonal coexistence of full non-determinism and probabilities without

distinguishing between states.

Segala and simple Segala probabilistic automata

Two types of probabilistic automata were introduced by Segala and Lynch in [SL94, Seg95b]. We call them Segala probabilistic automata and simple Segala probabilistic automata. An extensive overview of the simple Segala model is given in [Sto02a, Sto02b]. These types of probabilistic automata have been used for verification purposes and several theoretical results have been obtained as reported in [SV99, SV03, BS00, Bai96, BEMC99, BK00, BK97, JY02, KN98].

Definition 2.2.14. *A Segala probabilistic automaton is a triple* $\langle S, A, \alpha \rangle$ *where*

$$\alpha : S \to \mathcal{P}(\mathcal{D}(A \times S))$$

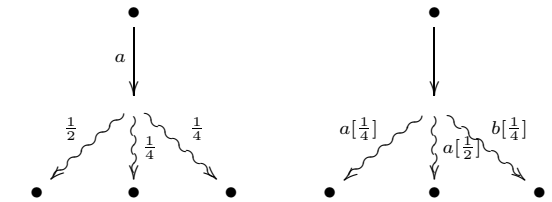
If $s \in S$ *and* $\mu \in \alpha(s)$ *we write* $s \to\!\!\!\!\rightsquigarrow \mu$*, and, if* $\langle a, s' \rangle \in \text{supp}(\mu)$ *with* $\mu(a, s') = p$ *we write* $s \xrightarrow{a[p]} s'$*.*

A simple Segala probabilistic automaton[1] *is a triple* $\langle S, A, \alpha \rangle$ *for a transition function*

$$\alpha : S \to \mathcal{P}(A \times \mathcal{D}(S))$$

If $s \in S$ *with* $\langle a, \mu \rangle \in \alpha(s)$ *then we write* $s \xrightarrow{a} \mu$*, and if* $s' \in \text{supp}(\mu)$ *we write* $s \xrightarrow{a\ p} s'$*.*

The simple Segala type of systems arise from **LTS** by changing the target state with a distribution over possible target states. A transition in a simple Segala automaton and in a Segala automaton is shown in the next figure.



simple Segala transition Segala transition

There can be more then one transition available in a state and that is where non-determinism occurs. Hence, non-deterministic choices exist between transitions, while probabilities are specified within a transition. In the original definition by Segala and Lynch distributions over an extended set $A \times S + 1$ (or over $S + 1$ in the simple case) were treated i.e. substochastic distributions, where the probability assigned to the special symbol $*$ was interpreted as the deadlock probability.

[1] Segala and Lynch call these models probabilistic automata (PA) and simple PA, while Stoelinga calls them general PA and PA, respectively.

We choose not to include this in the definition for two reasons: it disturbs the comparison (Section 2.4) since the other models do not cover substochastic distributions, and deadlock probability can be expressed by adding an extra deadlock state to a system.

We denote the class of Segala probabilistic automata by **Seg** and the class of simple Segala automata by **SSeg**.

The simple Segala automaton is a generalization towards full non-determinism of the reactive model and of the purely probabilistic automata of Rabin [Rab63]. A deterministic version of the simple Segala automaton equivalent to the reactive model is known as *Markov decision process* ([Der70]), while the name *probabilistic transition system* is used for this model in [JLY01] and for a state labelled version in [DJJL01, DJJL02]. A comparison of $\mathbf{SA}_n$ and the simple Segala model can be found in [BS01]. A very recent work provides another detailed comparison between alternating and non-alternating (Segala) models [ST05].

Bisimulation for the simple Segala systems is defined with the same transfer conditions as for reactive systems given in Definition 2.2.3, while for the Segala systems the transfer conditions for bisimulation of Definition 2.2.4 for generative systems apply, when changing $\rightsquigarrow$ to $\rightarrow\!\!\rightsquigarrow$. We state the definition for completeness.

Definition 2.2.15. *An equivalence relation R on S is a bisimulation on the Segala probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if for all $\langle s, t\rangle \in R$:*

if $s \rightarrow\!\!\rightsquigarrow \mu$, then there exists a distribution μ' with $t \rightarrow\!\!\rightsquigarrow \mu'$ and $\mu \equiv_{R,A} \mu'$.

A great novelty introduced with both types of Segala systems was the definition of a stronger probabilistic bisimulation relation that identifies states that have matching "combined transitions". For more information on this topic we refer to [SL94, Seg95b, Sto02a, Sto02b, BS00].

Bundle probabilistic automata

Another way to include both non-determinism and probability is to consider distributions over sets of transitions as in the *bundle* model, introduced in [DHK98]. (Recall that Segala systems have sets of distributions over transitions.)

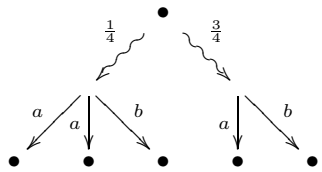
Definition 2.2.16. *A bundle probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where*

$$\alpha : S \rightarrow \mathcal{D}(\mathcal{P}(A \times S))$$

When $s \in S$ and $\alpha(s) = \mu$ we write $s \rightsquigarrow \mu$. Furthermore, if $T \subseteq A \times S$, $\mu(T) = p > 0$ we write $s \overset{p}{\rightsquigarrow} T$ and if $\langle a, t\rangle \in T$ then $s \overset{p}{\rightsquigarrow} \overset{a}{\rightarrow} t$.

Although not explicitly, a bundle automaton can also express termination. A terminating state $s \in S$ of a bundle automaton is characterized by a Dirac distribution transition $\alpha(s) = \mu^1_\emptyset$.

The bundle model can be considered as generative, since probabilities are also distributed over actions. Therefore, the bundle model offers a solution to the absence of non-determinism in the generative setting. Note that the original definition is even slightly more general, namely the codomain of the transition function is $\mathcal{D}(\mathcal{M}(A \times S))$ where $\mathcal{M}(X)$ denotes the collection of all the (finite) multi-subsets of a set X. Hence it is possible to have multiple transitions from one state to another with the same action within one bundle. Since it is not essential for the material presented here, we will not add multi-sets in the bundle model. The class of bundle probabilistic automata is denoted by **Bun**. A typical bundle probabilistic automaton is depicted below:



bundle probabilistic automaton

In the literature, in particular in [DHK98], there is no definition of bisimulation on bundle probabilistic automata, instead they are transformed to generative systems and then compared with generative bisimulation. We give here a definition of bisimulation for the bundle probabilistic automata that is deduced from the general coalgebraic definition of bisimulation (cf. Chapter 3 and Chapter 4).

Prior to stating the definition we need a way to lift a relation on a set S to a relation on the set $\mathcal{P}(A \times S)$.

Definition 2.2.17. *Let R be a relation on S and let $X,\ Y \in \mathcal{P}(A \times S)$. Define $X \equiv_{R,\mathcal{P}} Y$ if and only if for all $a \in A$:*

1. *if $\langle a, x\rangle \in X$, then there exists $\langle a, y\rangle \in Y$ with $\langle x, y\rangle \in R$*
2. *if $\langle a, y\rangle \in Y$, then there exists $\langle a, x\rangle \in X$ with $\langle x, y\rangle \in R$.*

It holds that, if R is an equivalence on S, then $\equiv_{R,\mathcal{P}}$ is an equivalence on $\mathcal{P}(A \times S)$.

Definition 2.2.18. *An equivalence relation R is a bisimulation on the state set of a bundle probabilistic automaton $\langle S, A, \alpha\rangle$ if and only if for all $\langle s, t\rangle \in R$ it holds*

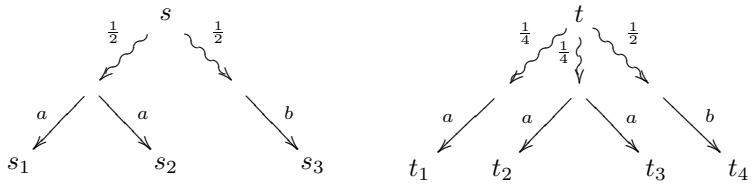
$$\text{if } s \rightsquigarrow \mu \text{ and } t \rightsquigarrow \mu',\ \text{then } \mu \equiv_{\equiv_{R,\mathcal{P}}} \mu'$$

where $\equiv_{\equiv_{R,\mathcal{P}}}$ denotes the lifting of the relation $\equiv_{R,\mathcal{P}}$ to distributions on $\mathcal{P}(A \times S)$ as defined by Definition 2.1.4.

It might seem that this bisimulation definition is different than all the others presented so far, but we shall see in later chapters (Chapter 3 and Chapter 4)

that they are all instantiations of a general definition of bisimulations on systems. To this end we present an example of bundle bisimulation.

Example 2.2.19. Let R be the equivalence on the set $S = \{s, t, s_1, s_2, s_3, t_1, t_2, t_3, t_4\}$ such that $S/R = \{\{s, t\}, \{s_1, s_2, s_3, t_1, t_2, t_3, t_4\}\}$. Consider the system:



The equivalence R is a bisimulation on this bundle system since the states $s_1, s_2, s_3, t_1, t_2, t_3, t_4$ are all terminating, and the distributions from the states s and t assign the same probability $1/2$ to the $\equiv_{R,\mathcal{P}}$-class containing the sets $\{\langle a, s_1\rangle, \langle a, s_2\rangle\}$, $\{\langle a, t_1\rangle\}$ and $\{\langle a, t_2\rangle, \langle a, t_3\rangle\}$, and the same probability $1/2$ to the $\equiv_{R,\mathcal{P}}$-class containing the sets $\{\langle b, s_3\rangle\}$ and $\{\langle b, t_4\rangle\}$.

2.2.4 Complex models - Pnueli-Zuck and general probabilistic automata

An early model including probabilities and a structured transition relation was proposed by Pnueli and Zuck [PZ86, PZ93] under the name *finite-state probabilistic programs* and later used in [BA95]. We call this type of automata Pnueli-Zuck probabilistic automata[2], and denote the class of all such by **PZ**. The model of Pnueli and Zuck has the most complex transition function of the models appearing in the literature, it adds one more power set to the bundle model and so allows two types of non-determinism, both between the probabilistic transitions and inside the transitions. However, in order to get a top element for our hierarchy (Section 2.4) we expand the model a bit further and define a most general type of probabilistic automata. The class of such will be denoted by **MG**.

Definition 2.2.20. *A Pnueli-Zuck automaton is a triple $\langle S, A, \alpha\rangle$ where*

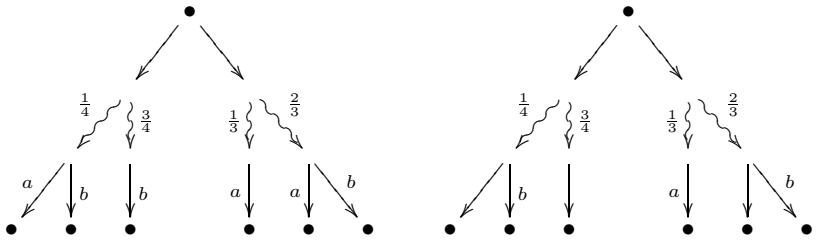
$$\alpha : S \to \mathcal{P}(\mathcal{D}(\mathcal{P}(A \times S)))$$

When $s \in S$ and $\mu \in \alpha(s)$ we write $s \to\!\!\!\!\rightsquigarrow \mu$. Furthermore, if $T \subseteq A \times S$, $\mu(T) = p > 0$ we write $s \overset{p}{\to\!\!\!\!\rightsquigarrow} T$ and if $\langle a, t\rangle \in T$ then $s \overset{p}{\to\!\!\!\!\rightsquigarrow} \overset{a}{\to} t$. A general probabilistic automaton is a triple $\langle S, A, \alpha\rangle$ where

$$\alpha : S \to \mathcal{P}(\mathcal{D}(\mathcal{P}(A \times S + S)))$$

[2]Like Vardi's model, these automata appear in the literature in a state labelled version for model checking purposes. Therefore, we change the definition towards transition labels.

The notation for Pnueli-Zuck automata is also used for general automata. Furthermore, if $s \in S, \mu \in \alpha(s), T \subseteq A \times S + S$ *with* $\mu(T) = p > 0$ *and* $t \in T \cap S$, *then we write* $s \xrightarrow{p} \rightarrow t$.



Pnueli-Zuck system most general system

The unlabelled transitions which appear in the right figure (most general system) correspond to pure probabilistic transitions in Markov chains or alternating systems, where a change of state can happen with certain probability without performing an action.

As for bundle systems, there is no notion of bisimulation for Pnueli-Zuck systems in the literature. A bisimulation definition can be formulated out of the general coalgebraic definition, and it leads the same transfer conditions as in Definition 2.2.18 when changing $\rightsquigarrow$ to $\rightarrow\rightsquigarrow$. A small modification is needed for the general probabilistic automata. Namely, the transfer condition from Definition 2.2.18 is still valid, but the definition of $\equiv_{R,\mathcal{P}}$ (Definition 2.2.17) has to be modified, such that besides the conditions 1. and 2. it also contains:

3. if $x \in X$, then there exists $y \in Y$ with $\langle x, y\rangle \in R$
4. if $y \in Y$, then there exists $x \in X$ with $\langle x, y\rangle \in R$.

2.3 Composing probabilistic systems in parallel

Having introduced the probabilistic models, we consider possible definitions of the parallel composition operator for these extended systems. In Section 2.1.3 we have discussed the importance and the different styles of parallel composition for LTS and for Markov chains. Now we focus on parallel composition of the various probabilistic automata. Lots of results on this topic exist in the literature. For a broad overview the reader is referred to [DHK98, Bai98]. An overview of probabilistic process algebras covering other probabilistic operators as well is presented in [LN04].

At this point we wish to emphasize the importance of the study of definability of parallel composition and its closure properties. A large part of the thesis is devoted to the study and evaluation of the various probabilistic systems. These systems are to be used as models of real systems for verification purposes. It

is often the case that models are constructed in a modular fashion, i.e. by building models of subsystems and composing them we get a model of the system. A rather important way of composing models of concurrent systems is via a parallel composition operator. We are interested in classes for which certain types of parallel composition operator can be defined. Moreover, if a class of automata is to be used for compositional modelling, then we want that the class is *closed* under the given parallel composition operator, so that when two automata of this type are composed in parallel the resulting automaton is still of the same type. In the next section and in Chapter 4 we discuss and build an expressiveness hierarchy of the types of probabilistic automata. We will see that more expressive and more complex models satisfy closure properties of the parallel composition operator.

For the classes **MC** and **IO** there is a unique parallel composition defined. In **MC** this operation is purely synchronous given by the product of distributions (cf. Section 2.1.3), whereas in **IO** the definition of the parallel composition operation strongly relies on the specific structure of the systems (cf. Section 2.3.3 below). For all other classes it is meaningful to consider various definitions of parallel composition. Such operations might be synchronous or asynchronous in nature and moreover might be based upon the styles CCS, CSP and ACP described in Section 2.1. The style CSP plays a special role in this respect since it is by its definition partly synchronous and partly asynchronous and hence gives rise to a somehow mixed variant of parallel composition.

The classes of (probabilistic) systems can be divided into three groups dependent on whether they show reactive, generative or alternating behavior. Classes belonging to the same of these groups allow in essence similar definition and investigation of parallel composition.

Before going through the, obviously quite numerous, variants of parallel composition, for each single class of systems in the subsequent Sections 2.3.1, 2.3.2 and 2.3.4, let us give a complete scheme of possible (and/or already studied) definitions of parallel composition operator by means of a comprehensive table. In the table each column is dedicated to one class of probabilistic automata, and each row to one of the introduced styles of parallel composition. In the intersecting cells a symbol representing the definability status of the corresponding parallel composition operator in the corresponding class is placed. Neighboring cells containing the same symbol within one column are merged. We use the following symbols:

- $\ast$: defined in the literature or straightforward
- $+$: definition possible but not carried out (here and/or in the literature)
- $-$: not definable
- P: defined in the literature with parameters
- p: parameterized version possible, but not carried out (here and/or lit.)
- n: normalized version possible, but not carried out (here and/or lit.)
- s_1/s_2: "s_1" for total communication function, "s_2" otherwise

		- - - reactive - - -				- - - - - generative - - - - -						- - - alternating - - -	
		DLTS	**LTS**	**React**	**SSeg**	**Gen**	**Seg**	**Bun**	**PZ**	**MG**	**Var**	**Alt, SA, ..**	**Str**
sync	CCS	–		–		n	n				n		
	CSP	*	*	*	*	n	n	*	+	+	n	*	*
	ACP	–		–		*/n	+/n				+/n		
	CSP	–	*	–	*	P	p	*	+	+	p	*	–
async	CCS												
	CSP	–	*	–	*	P	p	*	+	+	p	*	–
	ACP												

Table 2.1: Definability of $\|$

In the above explanation of symbols s_1 and s_2 may be any of the previously introduced symbols. Table 2.1 presents the overall situation concerning definability of parallel composition on probabilistic automata.

Hence, for example, from the table we see that for generative systems a synchronous ACP parallel composition operator can be defined provided that the communication function is a total function. Otherwise, a normalized version of parallel composition can be defined (symbol "∗/n").

A brief analysis of these summary results shows that allowing full nondeterminism enables definition of any type of parallel composition. In the remainder of this section we discuss the definitions of parallel composition for all of the mentioned systems and we point out to the relevant literature.

2.3.1 Parallel composition in the reactive setting

Systems with reactive behavior are systems in the classes **React** and **SSeg**, as well as **LTS** and **DLTS** in the non-probabilistic case. Any parallel composition operator on LTS (Section 2.1.3) nicely extends to the class **SSeg**. Let $\mathcal{A}_1 = \langle S_1, A, \alpha_1 \rangle, \mathcal{A}_2 = \langle S_2, A, \alpha_2 \rangle$ be in **SSeg**. Then $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S_1 \times S_2, A, \alpha \rangle$ where α is defined as follows:

[*CCS style*]: $s \| t \xrightarrow{a} \mu$ if and only if

1. $a = \tau$, $s \xrightarrow{b} \mu_1$, $t \xrightarrow{\bar{b}} \mu_2$ and $\mu = \mu_1 \times \mu_2$, or
2. $s \xrightarrow{a} \mu_1$ and $\mu = \mu_1 \times \mu_t^1$, or

3. $t \xrightarrow{a}\!\!\rightsquigarrow \mu_2$ and $\mu = \mu_s^1 \times \mu_2$.

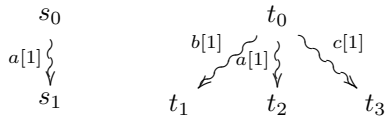
[*CSP style*]: $s\|t \xrightarrow{a}\!\!\rightsquigarrow \mu$ if and only if

1. $a \in L$, $s \xrightarrow{a}\!\!\rightsquigarrow \mu_1$, $t \xrightarrow{a}\!\!\rightsquigarrow \mu_2$ and $\mu = \mu_1 \times \mu_2$, or
2. $a \notin L$, $s \xrightarrow{a}\!\!\rightsquigarrow \mu_1$ and $\mu = \mu_1 \times \mu_t^1$, or
3. $a \notin L$, $t \xrightarrow{a}\!\!\rightsquigarrow \mu_2$ and $\mu = \mu_s^1 \times \mu_2$.

[*ACP style*]: $s\|t \xrightarrow{a}\!\!\rightsquigarrow \mu$ if and only if

1. $a = bc$ defined, $s \xrightarrow{b}\!\!\rightsquigarrow \mu_1$, $t \xrightarrow{c}\!\!\rightsquigarrow \mu_2$ and $\mu = \mu_1 \times \mu_2$, or
2. $s \xrightarrow{a}\!\!\rightsquigarrow \mu_1$ and $\mu = \mu_1 \times \mu_t^1$, or
3. $t \xrightarrow{a}\!\!\rightsquigarrow \mu_2$ and $\mu = \mu_s^1 \times \mu_2$.

The definition of any of these operators is problematic for the class **React**. For $\mathcal{A}_1, \mathcal{A}_2 \in$ **React** it might happen that $\mathcal{A}_1 \| \mathcal{A}_2 \notin$ **React** in any variant of parallel composition. Even in the synchronous CCS style, multiple transitions labelled with τ may appear. In the CSP style, 2. and 3. may introduce internal non-determinism. However, if L contains all the common actions of $\mathcal{A}_1$ and $\mathcal{A}_2$, then this problem disappears. In case of ACP all of 1., 2. and 3. introduce internal non-determinism, hence **React** is not closed under this operator for an arbitrary communication function γ. The same problems arise in the class **DLTS**, namely parallel composition introduces internal non-determinism, and therefore **DLTS** is not closed under $\|$. For example, let γ be the smallest commutative and associative partial function on the set of actions $A = \{a, b, c\}$ such that $\gamma(a, b) = ab = a$ and $\gamma(a, c) = ac = a$. It is easy to see that such a partial function exists. Then the ACP parallel product of the following two automata



is not defined in **React**, since the definition yields: $s_0\|t_0 \xrightarrow{a}\!\!\rightsquigarrow \mu_x^1$ for $x \in \{s_1\|t_0, s_0\|t_2, s_1\|t_1, s_1\|t_3\}$ i.e. more than one transition corresponds to the action a, which is prohibited in **React**.

An asynchronous parallel composition in CCS style on simple Segala systems was defined in [BK00], a synchronous parallel composition in CCS/ACP[3] style

[3]The authors refer to this synchronous parallel composition as (S)CCS-style. With the notation introduced so far, it is in fact an ACP style parallel composition where the set of labels is a free semigroup A^* and for $u, v \in A^*$ the communication function is defined as $\gamma(u, v) = uv$.

on reactive systems was defined in [GSST90, GSS95, JLY01], the last reference working with simple Segala systems. A synchronous CSP style parallel composition is defined for reactive systems in [JY02, Nor97], while an asynchronous CSP style parallel composition with synchronization on common actions is used in [SL94, Seg95b, Sto02a] for simple Segala systems.

2.3.2 Parallel composition in the generative setting

Systems with generative behavior belong to the classes **Gen**, **Var**, **Seg**, **Bun**, **PZ** and **MG**. The Vardi systems express also alternating behavior and they will be discussed with the alternating systems. A common property of the generative systems is that always probability distributions over actions and states appear. This leads to difficulties in defining parallel composition operators (see [Han91, CSZ92, Seg95b, DHK98]), especially in the asynchronous case. Namely, a generative type system defines in each state a probability distribution over a set of enabled actions, offered by the environment. When two such systems are composed in parallel it is not clear how the common set of enabled actions should be defined, nor how the two probability distributions should be composed into one (cf. [JLY01]). In this section we explain several approaches for solving this problem.

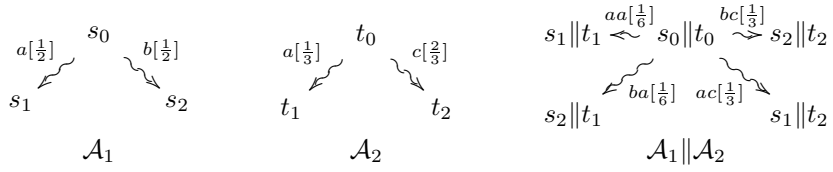
Let $\mathcal{A}_1 = \langle S_1, A, \alpha_1 \rangle, \mathcal{A}_2 = \langle S_2, A, \alpha_2 \rangle$ be two generative systems. Their parallel composition in all cases will be denoted by $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S_1 \times S_2, A, \alpha \rangle$, possibly with parameters.

Synchronous CCS, CSP, ACP style parallel composition can be defined on generative systems, as done in [GSST90, GSS95, DHK98] by

$$s \overset{a[p]}{\rightsquigarrow} s', t \overset{b[q]}{\rightsquigarrow} t' \iff s \| t \overset{ab[pq]}{\rightsquigarrow} s' \| t'$$

where the set of actions is assumed to form a commutative semigroup (PCCS, [GJS90]) and ab stands for the product of a and b in $A(\cdot)$.

The following figure presents an example of synchronous parallel composition of two generative systems.



In order to capture possible asynchronous behavior, several parallel composition operators were defined in the literature that use *bias factors*. In most of the cases the composition is not symmetric. Namely, the main problem in defining asynchronous parallel composition is that any definition introduces non-determinism. In the proposed solutions, these non-deterministic choices are

changed to probabilistic ones, by specifying parameters of the parallel composition.

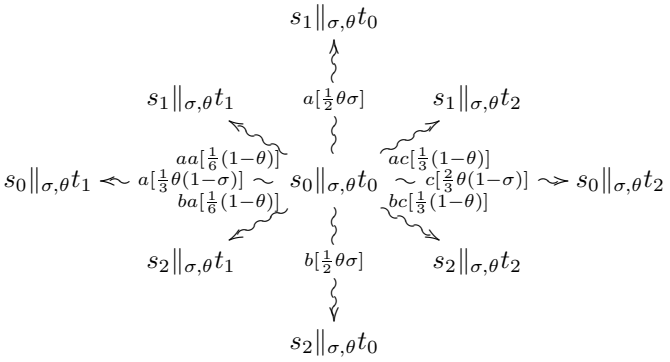
An ACP style parallel composition operator for generative systems was defined in [BBS95]. The definition follows the non-probabilistic definition of the ACP parallel operator, while changing the non-deterministic choices introduced by interleaving and/or communication into corresponding probabilistic choices. The operator is parameterized with two parameters σ and θ, $0 \leq \sigma, \theta \leq 1$, denoted by $\mathcal{A}_1 \|_{\sigma,\theta} \mathcal{A}_2$. In the product state $s\|_{\sigma,\theta}t$, synchronization between s and t can occur with probability $1-\theta$ and an autonomous action of either s or t with probability θ. Furthermore, given that an autonomous move occurs, then it comes from s with probability σ and from t with probability $1-\sigma$. For definability of $\mathcal{A}_1 \|_{\sigma,\theta} \mathcal{A}_2$ it is necessary that the communication function is a total function.

We define $s\|_{\sigma,\theta}t \overset{a[P]}{\rightsquigarrow} s'\|_{\sigma,\theta}t'$ if and only if

1. $s \overset{b[p]}{\rightsquigarrow} s'$ and $t \overset{c[q]}{\rightsquigarrow} t'$, $bc = a$ and $P = (1-\theta)pq$, or
2. $s \overset{a[p]}{\rightsquigarrow} s'$, $t = t'$, and $P = p\theta\sigma$, or
3. $t \overset{a[q]}{\rightsquigarrow} t'$, $s = s'$ and $P = q\theta(1-\sigma)$, or
4. $s \overset{a[p]}{\rightsquigarrow} s'$, $t \not\rightsquigarrow$, $t = t'$ and $P = p$, or
5. $s \not\rightsquigarrow$, $t \overset{a[q]}{\rightsquigarrow} t'$, $s = s'$ and $P = q$.

Note that by this definition we might get two transitions $s\|_{\sigma,\theta}t \overset{a[p_1]}{\rightsquigarrow} s'\|_{\sigma,\theta}t'$ and $s\|_{\sigma,\theta}t \overset{a[p_2]}{\rightsquigarrow} s'\|_{\sigma,\theta}t'$, which then are replaced by one transition $s\|_{\sigma,\theta}t \overset{a[p_1+p_2]}{\rightsquigarrow} s'\|_{\sigma,\theta}t'$.

For $\mathcal{A}_1$ and $\mathcal{A}_2$ as in the previous figure, we get $\mathcal{A}_1 \|_{\sigma,\theta} \mathcal{A}_2$ which looks like:



Two other biased, parameterized, parallel composition operators are defined in [DHK98], one asynchronous CCS style operator, denoted by $\mathcal{A}_1^{\theta} \|^{\sigma} \mathcal{A}_2$ and

one CSP-style operator, for a set of synchronizing labels $L \subseteq A$, denoted by $\mathcal{A}_1 \|^\sigma \mathcal{A}_2$. Denote by $s \overset{a[p]}{\rightsquigarrow}$ the clause $(\exists s')s \overset{a[p]}{\rightsquigarrow} s'$. We present the definition of $\mathcal{A}_1 \|^\sigma \mathcal{A}_2$ first:

$s\|^\sigma t \overset{a[P]}{\rightsquigarrow} s'\|^\sigma t'$ if and only if one of the following is satisfied:

1. $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{b[q]}{\rightsquigarrow}$, $a, b \notin L$, $t' = t$ and $P = \frac{pq\sigma}{\nu(s,t,L)}$
2. $s \overset{b[p]}{\rightsquigarrow}$, $t \overset{a[q]}{\rightsquigarrow} t'$, $a, b \notin L$, $s' = s$ and $P = \frac{pq(1-\sigma)}{\nu(s,t,L)}$
3. $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{b[q]}{\rightsquigarrow}$, $a \notin L, b \in L$, $t' = t$ and $P = \frac{pq}{\nu(s,t,L)}$
4. $s \overset{b[p]}{\rightsquigarrow}$, $t \overset{a[q]}{\rightsquigarrow} t'$, $a \notin L, b \in L$, $s' = s$ and $P = \frac{pq}{\nu(s,t,L)}$
5. $s \overset{a[p]}{\rightsquigarrow} s'$, $t \not\rightsquigarrow$, $a \notin L$, $t' = t$ and $P = \frac{p}{\nu'(s,L)}$
6. $s \not\rightsquigarrow$, $t \overset{a[q]}{\rightsquigarrow} t'$, $a \notin L$, $s' = s$ and $P = \frac{q}{\nu'(t,L)}$
7. $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{a[q]}{\rightsquigarrow} t'$, $a \in L$ and $P = \frac{pq}{\nu(s,t,L)}$.

Where the normalization factors are calculated by

$$\nu'(s, L) = 1 - \sum_{s \overset{a[p]}{\rightsquigarrow},\ a \in L} p,\ \nu(s, t, L) = 1 - \sum_{s \overset{a[p]}{\rightsquigarrow},\ t \overset{b[q]}{\rightsquigarrow},\ a,b \in L,\ a \neq b} pq.$$

For this CSP style operator only one parameter is needed since the only non-determinism occurs if both systems autonomously decide to perform actions not in the synchronizing set L. In $s\|^\sigma t$, the parameter σ denotes the probability that s performs an autonomous action, given that both s and t have decided not to synchronize. Furthermore, normalization factors are used to determine the actual probability of every transition. These normalization factors redistribute the probability mass that is due to autonomous decisions of both processes that would otherwise lead to deadlock.

For the asynchronous CCS parallel composition $\mathcal{A}_1 \,{}^\theta\|^\sigma \mathcal{A}_2$ the interpretation of the probabilistic parameters $\theta, \sigma \in (0, 1)$ is similar to the ACP approach. They provide the relevant information that an adversary needs in order to resolve non-determinism that arises when composing two systems. In $s \,{}^\theta\|^\sigma t$, σ denotes the probability that s performs an autonomous action given that both s and t do not want to synchronize, and θ denotes the probability that some autonomous action occurs, given that synchronization is possible. Hence, if synchronization is possible, it will take place with probability $1 - \theta$.

The earliest biased parallel composition operator for generative systems was defined in [CSZ92] and discussed in detail in [LN04]. There the parallel composition $\mathcal{A}_1\|_\rho \mathcal{A}_2 = \langle S_1 \times S_2, A, \alpha \rangle$ uses one bias parameter ρ. A state $s\|_\rho t$ in the

composed automaton can either do an action a with a certain probability if both s and t in their components can do an action a (CSP style), or can do a τ action if either s or t can do a τ action. However, whether a τ action from s or from t is chosen is biased with the bias factor ρ. The probability of the synchronous execution of a is calculated via a normalization function $\nu : S_1 \times S_2 \to [0,1]$. Basically, $\nu(s,t)$ sums up the probabilities of the possible outgoing transitions of the new state which would be obtained if asynchronous behavior introduced non-determinism. Then $\nu(s,t)$ is used to calculate the actual (conditional) probabilities of the distribution assigned to $s\|_\rho\, t$.

Finally, a completely different solution of the problem of defining a parallel composition operator in the generative setting is provided in [DHK98], the introduction of the class of bundle systems **Bun**. The bundle systems possess non-determinism, which allows for an elegant definition of an asynchronous parallel composition operator, as follows.

Let $\mathcal{A}_1 = \langle\, S_1, A, \alpha_1 \,\rangle, \mathcal{A}_2 = \langle\, S_2, A, \alpha_2 \,\rangle \in \mathbf{Bun}$. Then $\mathcal{A}_1 \| \mathcal{A}_2 = \langle\, S_1 \times S_2, A, \alpha \,\rangle$ where, for $s \in S_1, t \in S_2$,

$$s\|t \rightsquigarrow \mu = P(\mu_1, \mu_2) \iff s \rightsquigarrow \mu_1, t \rightsquigarrow \mu_2$$

and $P(\mu_1, \mu_2)$ denotes a specific product of distributions, defined as follows: For $\mu_1 \in \mathcal{D}(\mathcal{P}(A \times S_1)), \mu_2 \in \mathcal{D}(\mathcal{P}(A \times S_2))$, $\mu = P(\mu_1, \mu_2) \in \mathcal{D}(\mathcal{P}(A \times (S_1 \times S_2)))$ where for all $B_s \in \mathrm{supp}(\mu_1), B_t \in \mathrm{supp}(\mu_2)$, $\mu(B_s \otimes B_t) = \mu_1(B_s) \cdot \mu_2(B_t)$ and

$$\begin{aligned} &B_s \otimes B_t = \\ &\{\langle\, a, \langle\, s', t \,\rangle\,\rangle \mid \langle\, a, s' \,\rangle \in B_s\} \cup \\ &\{\langle\, b, \langle\, s, t' \,\rangle\,\rangle \mid \langle\, b, t' \,\rangle \in B_t\} \cup \\ &\{\langle\, ab, \langle\, s', t' \,\rangle\,\rangle \mid \langle\, a, s' \,\rangle \in B_s, \langle\, b, t' \,\rangle \in B_t\}. \end{aligned}$$

Note that the defined parallel composition for bundle systems is ACP style. By a slight modification of the definition of $\otimes$, all the other variants can be obtained. In a similar manner asynchronous parallel composition can be defined on the classes **PZ** and **MG**. In the literature there is no definition of a parallel composition operator for the class **Seg**.

2.3.3 Parallel composition in the I/O setting

A rather clean solution to the problems in the generative setting is given for the class of I/O automata in [WSS97]. The view taken there is that the actions are divided into input and output, and while there can be synchronization on input actions, as in the reactive setting, the sets of output actions in each of the components must be disjoint. This style of parallel composition is also found in the process algebra EMPA [BG98, Ber99].

Let $\mathcal{A}_1 = \langle S_1, A_1, \alpha_1 \rangle, \mathcal{A}_2 = \langle S_2, A_2, \alpha_2 \rangle$ be two I/O automata. The automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are *compatible* if and only if $A_1^{out} \cap A_2^{out} = \emptyset$. Parallel composition is

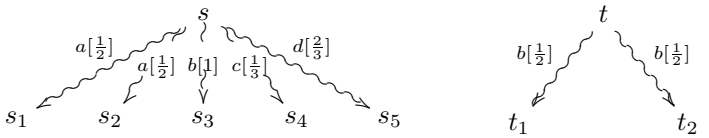
only defined on compatible automata. Let $A = A_1 \cup A_2$. We use the following convention: if s is a state in $\mathcal{A}_1$ ($\mathcal{A}_2$) and $a \in A \setminus A_1^{out}$ ($A \setminus A_2^{out}$) is such that there is no transition from s involving a, then we consider that $s \overset{a[1]}{\rightsquigarrow} s$. This convention will enforce the "input always enabled" requirement for the composite automaton.

The parallel composition of $\mathcal{A}_1$ and $\mathcal{A}_2$ is the I/O automaton $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S_1 \times S_2, A, \alpha \rangle$ where:

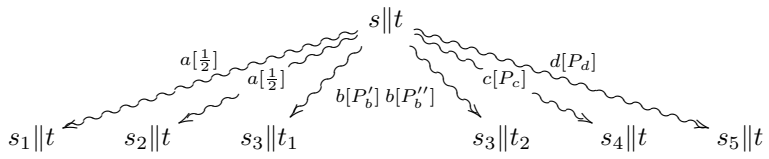
1. $A^{out} = A_1^{out} \cup A_2^{out}$
2. $A^{in} = A \setminus A^{out} = (A_1^{in} \cup A_2^{in}) \setminus A^{out}$
3. the transition function α is defined by the following:
   $s\|t \overset{a[P]}{\rightsquigarrow} s'\|t'$ if and only if one of the following holds
   a. $a \in A^{in}$, $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{a[q]}{\rightsquigarrow} t'$ and $P = pq$
   b. $a \in A_1^{out}$, $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{a[q]}{\rightsquigarrow} t'$ and $P = \frac{\delta_1(s)}{\delta_1(s)+\delta_2(t)} pq$
   c. $a \in A_2^{out}$, $s \overset{a[p]}{\rightsquigarrow} s'$, $t \overset{a[q]}{\rightsquigarrow} t'$ and $P = \frac{\delta_2(t)}{\delta_1(s)+\delta_2(t)} pq$

   Hence, $\alpha(s\|t) = \langle f^{in}, \mu^{out}, \delta \rangle$ where $\delta(s\|t) = \delta_1(s) + \delta_2(t)$, $f^{in}(a) = \mu_a$ is determined by clause 3a., for $a \in A^{in}$, and μ^{out} is determined by 3b. and 3c., for $a \in A^{out}$.

Example 2.3.1. Let $\mathcal{A}_1 = \langle S_1, A_1, \alpha_1 \rangle, \mathcal{A}_2 = \langle S_2, A_2, \alpha_2 \rangle$ be two I/O automata, with $s \in S_1, t \in S_2$ and their corresponding transitions as in the following diagram.



Take $A_1^{in} \supseteq \{a, b\}$, $A_1^{out} \supseteq \{c, d\}$, A_2^{in} any set, $A_2^{out} \supseteq \{b\}$. Assume the two automata are compatible i.e. $A_1^{out} \cap A_2^{out} = \emptyset$ (clearly the states s and t are compatible). Then $A^{out} \supseteq \{b, c, d\}$ and $A^{in} \supseteq \{a\}$. Due to the convention we consider that $t \overset{a[1]}{\rightsquigarrow} t$, $t \overset{c[1]}{\rightsquigarrow} t$ and $t \overset{d[1]}{\rightsquigarrow} t$. The transitions from $s\|t$ are then given with the following diagram.



For $P'_b = P''_b = \frac{\delta_2(t)}{\delta_1(s)+\delta_2(t)} \cdot \frac{1}{2}$, $P_c = \frac{\delta_1(s)}{\delta_1(s)+\delta_2(t)} \cdot \frac{1}{3}$ and $P_d = \frac{\delta_1(s)}{\delta_1(s)+\delta_2(t)} \cdot \frac{2}{3}$. Note that indeed $P'_b + P''_b + P_c + P_d = 1$.

The proof that $\mathcal{A}_1 \| \mathcal{A}_2$ is well defined in the class **IO** can be found in [WSS97]. Let us now informally explain the definition of parallel composition, and the role of the functions δ_1, δ_2 and δ. If s is a state of $\mathcal{A}_1$, then $\delta_1(s)$ is a positive real number corresponding to the delay rate in state s. It is a rate of an exponential distribution, determining the time that the automaton waits in state s until it generates one of its output actions. If no output actions are enabled in this state then $\delta_1(s) = 0$. When determining the distribution on output actions for $s \| t$, denoted by $\mu^{out}_{s\|t}$, the components' distributions μ^{out}_s and μ^{out}_t are joined in one such that any probability of μ^{out}_s is multiplied with normalization factor $\frac{\delta_1(s)}{\delta_1(s)+\delta_2(t)}$ and any probability of μ^{out}_t is multiplied with $\frac{\delta_2(t)}{\delta_1(s)+\delta_2(t)}$. Note that by the compatibility assumption, no action appears both in the support of μ^{out}_s and in the support of μ^{out}_t. The normalization factor models a racing policy between the states s and t for generating their own output actions. The value $\frac{\delta_1(s)}{\delta_1(s)+\delta_2(t)}$ is the probability that the state s has less waiting time left than the state t and therefore wins the race and generates one of its own output actions. On the other hand, synchronization occurs on all input actions, no autonomous behavior is allowed by the components on input actions, corresponding to the assumption that the input is provided by the environment and must be enabled in any state.

A new model with I/O distinction has been proposed recently [CH05] in order to achieve compositionality of trace-style semantics for I/O automata. The model is designed so that it is closed with respect to a meaningful parallel composition operator in the I/O style. This shows that indeed the need of compositionality and closure properties motivates investigating new models.

2.3.4 Parallel composition in the alternating setting

In this section we focus on the classes **Str**, **Alt**, **SA** ($\mathbf{SA}_n$, $\mathbf{SA}_p$) and **Var** that exhibit alternating behavior i.e. make a distinction between probabilistic and non-deterministic states. In [GSST90, GSS95] and in [Han91] a rather elegant parallel composition for the classes **Str** and **SA**, respectively, is defined.

We present the definition for the class **Alt** and discuss that the same definition can be restricted to the classes **Str**, **Alt**, **SA** ($\mathbf{SA}_n$, $\mathbf{SA}_p$). Let $\mathcal{A}_1 = \langle S_1, A, \alpha_1 \rangle$ and $\mathcal{A}_2 = \langle S_2, A, \alpha_2 \rangle$ be two alternating automata, with $S_1 = N_1 + P_1$ and $S_2 = N_2 + P_2$. Their parallel composition is the alternating automaton $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S, A, \alpha \rangle$ where $S = S_1 \times S_2 = N + P$ for $N = N_1 \times N_2$ and $P = P_1 \times P_2 + N_1 \times P_2 + P_1 \times N_2$ and the transition function is defined as follows. Let $p_1 \in P_1, p_2 \in P_2, n_1 \in N_1, n_2 \in N_2$ and $s_1 \in S_1, s_2 \in S_2$. For the probabilistic states in the composed automaton, we have:

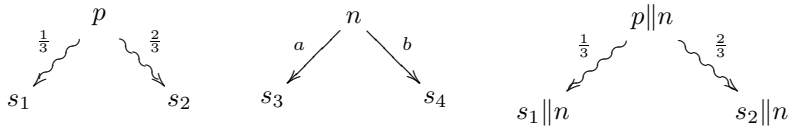
$$
\begin{array}{rcl}
p_1 \| p_2 \rightsquigarrow \mu & \iff & p_1 \rightsquigarrow \mu_1, p_2 \rightsquigarrow \mu_2, \mu = \mu_1 \times \mu_2 \\
p_1 \| n_2 \rightsquigarrow \mu & \iff & p_1 \rightsquigarrow \mu_1, \mu = \mu_1 \times \mu^1_{n_2} \\
n_1 \| p_2 \rightsquigarrow \mu & \iff & p_2 \rightsquigarrow \mu_2, \mu = \mu^1_{n_1} \times \mu_2
\end{array}
$$

where $\mu^1_{n_2}$ denotes the Dirac distribution for the state n_2.

For the non-deterministic states in the composed automaton different variants (CCS, CSP or ACP style) can be chosen. We choose for the ACP style:
$n_1 \| n_2 \xrightarrow{a} s_1 \| s_2$ if and only if

1. $n_1 \xrightarrow{b} s_1$, $n_2 \xrightarrow{c} s_2$ and $bc = a$ defined, or
2. $n_1 \xrightarrow{a} s_1$, $n_2 = s_2$, or
3. $n_2 \xrightarrow{a} s_2$, $n_1 = s_1$.

Hence, when composing a probabilistic state with any other state the result is a probabilistic state. If the other state is non-deterministic, then the composed state basically behaves as the probabilistic state and leaves the second component of the state unchanged, as in the following example.



On the other hand, the composition of non-deterministic states is exactly the same as in the non-probabilistic case and therefore all probabilistic counterparts of LTS parallel composition operators are definable here as in the case for LTSs.

By inspecting the definitions of the classes **SA**, $\mathbf{SA}_n$ and $\mathbf{SA}_p$ it is easy to see that the following statement is valid.

Proposition 2.3.2. *If $\mathcal{A}_1, \mathcal{A}_2 \in \mathbf{SA}$ (or $\mathbf{SA}_n$, or $\mathbf{SA}_p$), then $\mathcal{A}_1 \| \mathcal{A}_2 \in \mathbf{SA}$ (or $\mathbf{SA}_n$, or $\mathbf{SA}_p$, respectively).* □

The definition of parallel composition for stratified systems is given in [GSST90, GSS95] with synchronous behavior when composing two non-deterministic states. This is necessary in order to stay in the class **Str** when composing two such automata, since in the stratified model there is only a single action transition possible from a (non-) deterministic state. A parallel composition operator with no synchronization but only interleaving, for the stratified class of systems, is defined in [HV98]. In the original definition for strictly alternating systems of [Han91], non-deterministic states are composed in the CCS fashion.

Complications arise in the case of **Var** models, due to their generative probabilistic behavior. The behavior of the composite states $n_1 \| n_2$, $n_1 \| p_2$ and $p_1 \| n_2$ can be defined in the same way as above. However, there is no convenient way to

define $p_1 \| p_2$, since this coincides with defining generative parallel composition. Any of the approaches described in Section 2.3.2 can be used.

We mention that a different (asynchronous, ACP style) parallel composition operator is defined on alternating models in a process algebraic setting by Andova [And02] leading to finite axiomatization of the parallel composition operator.

2.4 Comparing classes

In the previous sections we defined the probabilistic models and their bisimulations, and we considered the definability of various parallel composition operators on different types of probabilistic systems in the context of closure properties. This section focuses on the inter-relationships between the various probabilistic models. We start by stating a comparison criterion for relative expressiveness of one class of automata with respect to another class, and then we present a hierarchy of the classes. The main results from this section will be proved in Chapter 4, using the unifying power of the theory of coalgebras (Chapter 3). However, the results can be stated, motivated and explained with the machinery introduced so far.

An expressiveness criterion

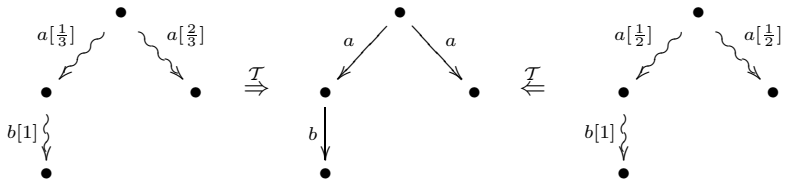
Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be two classes of probabilistic automata. We say that the class $\mathbf{C}_1$ is *included* or *embedded* in the class $\mathbf{C}_2$, i.e. the class $\mathbf{C}_2$ is *at least as expressive as* the class $\mathbf{C}_1$ (notation $\mathbf{C}_1 \to \mathbf{C}_2$) if and only if there exists an injective function $\mathcal{T}$ that maps each automaton of the first class to an automaton of the second class such that bisimilarity is both reflected and preserved. More explicitly, the translation function $\mathcal{T} : \mathbf{C}_1 \to \mathbf{C}_2$ should satisfy:

1. for $\mathcal{A} = \langle S, A, \alpha \rangle$ in $\mathbf{C}_1$, $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ with the same set of states S
2. the translation function $\mathcal{T}$ is injective, and
3. if $s, t \in S$, then $s \sim_{\mathcal{A}} t \Leftrightarrow s \sim_{\mathcal{T}(\mathcal{A})} t$, i.e. two states are bisimilar in the translated automaton (according to bisimilarity in the class $\mathbf{C}_2$) if and only if they were bisimilar in the original automaton (according to bisimilarity for the class $\mathbf{C}_1$).

The relation $\to$ between the classes of (probabilistic) automata is clearly a preorder.

Basically, our expressiveness criterion states that the class $\mathbf{C}_1$ is really embedded in the class $\mathbf{C}_2$, i.e. the translations are nothing else but "suitable copies" of the automata of the first class existing in the second class. Note that only

preservation of bisimulation is not enough. For example, we could define a translation from reactive systems to LTSs that preserves bisimulation, by forgetting the probabilities, as in the following example.



But we do not consider the class of LTSs more expressive than the class of reactive probabilistic automata, and the translation is by no means injective.

A similar expressiveness criterion was used for expressiveness comparison of timed languages by Cacciagrano and Corradini [CC04]. Another hierarchy result is the hierarchy of reactive, generative, stratified and non-deterministic automata of [GSST90] and [GSS95]. The expressiveness criterion used by Van Glabbeek et al. is different. They consider a class $\mathbf{C}_2$ at least as expressive as a class $\mathbf{C}_1$ if there is an *abstraction* mapping that maps each automaton of the class $\mathbf{C}_2$ to an automaton of the class $\mathbf{C}_1$, such that the state set remains the same and bisimulation is preserved. The abstraction mappings are not injective by nature. An example of such an abstraction mapping is the translation that forgets the probabilities mentioned above. Therefore, in their setting the class **React** is proved at least as expressive as the class **LTS**.

The hierarchy

Theorem 2.4.1. *[BSV03] The class embeddings presented in Figure 2.1 hold among the probabilistic system types.* □

The proof of Theorem 2.4.1 (except for the strictly alternating classes) is a subject of Chapter 4 below, using elements of the theory of coalgebras. The coalgebraic framework proves to provide the necessary abstraction for an elegant and interesting proof. Still, here we will explicitly state the translations for each arrow in Figure 2.1, give some examples and illustrate how preservation and reflection of bisimulation can be proved in one concrete case. We present the translations for the arrows of Figure 2.1 in several groups: simple arrows based on inclusion, arrows that show a change from external to full non-determinism, arrows that change an element to a singleton, arrows that change an element to a corresponding Dirac distribution and more specific arrows. The translations are quite natural. It is the property of preservation and reflection of bisimilarity that adds justification to the translations.

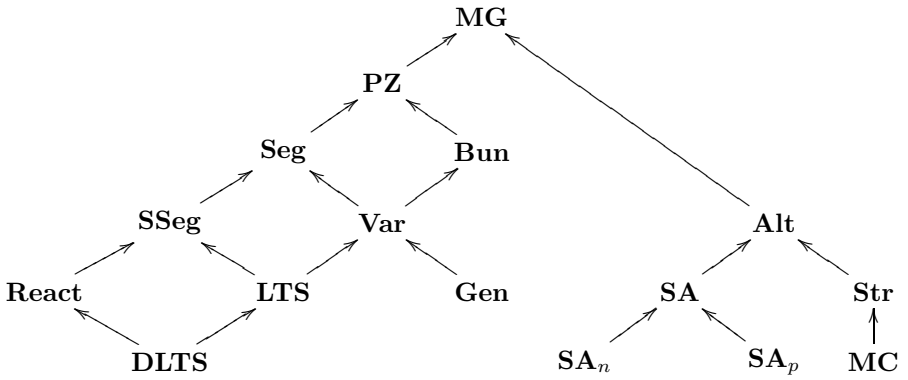


Figure 2.1: Class Embeddings

Simple arrows

Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be two classes of probabilistic automata. If $\mathbf{C}_1 \subseteq \mathbf{C}_2$ then $\mathbf{C}_1 \to \mathbf{C}_2$. Therefore, the embeddings $\mathbf{SA} \to \mathbf{Alt}$, $\mathbf{SA}_n \to \mathbf{SA}$ and $\mathbf{SA}_p \to \mathbf{SA}$ hold. Furthermore, if $\mathbf{C}_1$ is defined with a transition function $\alpha_1 : S \to C_1(S)$ and $\mathbf{C}_2$ has a transition function of type $\alpha_2 : S \to C_2(S)$ such that for all S, $C_1(S) \subseteq C_2(S)$ then every automaton of the class $\mathbf{C}_1$ can be considered an automaton of the class $\mathbf{C}_2$, by only extending the codomain of the transition function. In this case also $\mathbf{C}_1 \to \mathbf{C}_2$. The following arrows of Figure 2.1 hold due to extending the codomain of the transition function: $\mathbf{MC} \to \mathbf{Str}$, $\mathbf{Gen} \to \mathbf{Var}$, $\mathbf{LTS} \to \mathbf{Var}$, $\mathbf{PZ} \to \mathbf{MG}$. In each of these cases the translation of the automata is basically the identity mapping. For example, every generative automaton is a Vardi automaton without non-deterministic states, or every Markov chain is a stratified automaton that has no action-transitions, i.e. no deterministic states.

From external to full non-determinism

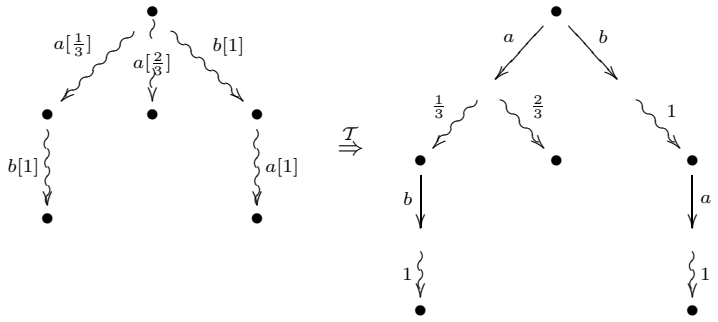
Two of the embedding arrows of Figure 2.1, $\mathbf{DLTS} \to \mathbf{LTS}$ and $\mathbf{React} \to \mathbf{SSeg}$, show that every system with only external non-determinism can be considered as a system with full non-determinism that never uses the full-nondeterminism option. Let $\mathcal{A} = \langle S, A, \alpha \rangle \in \mathbf{DLTS}$. Then $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle \in \mathbf{LTS}$ is given by

$$\alpha'(s) = \{\langle a, s' \rangle \mid \alpha(s)(a) = s' \in S\} \in \mathcal{P}(A \times S).$$

If we consider automata as diagrams, i.e. transition graphs, then this translation does not change the transition graph of a **DLTS** system.

For $\mathbf{React} \to \mathbf{SSeg}$ a similar translation is used, changing a partial function to its graph. Due to the notation used for reactive systems in Section 2.2, the

diagram of a reactive system when considered a simple Segala system has to be re-drawn, as in the next example.

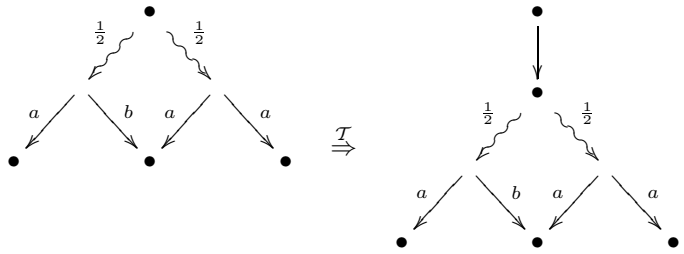


a reactive system as a simple Segala system

Singleton arrows

In several cases the translation only changes an element (state, or pair of state and action, or distribution) into a singleton set containing this element.

Bun $\rightarrow$ **PZ**: Let $\mathcal{A} = \langle S, A, \alpha \rangle$ be a bundle probabilistic automaton, i.e. $\alpha : S \rightarrow \mathcal{D}(\mathcal{P}(A \times S))$, then the translation to a Pnueli-Zuck automaton is achieved by putting $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ for $\alpha'(s) = \{\alpha(s)\}$.



a bundle transition as a Pnueli-Zuck transition

Str $\rightarrow$ **Alt**: In this case $\mathcal{T}(\langle S, A, \alpha \rangle) = \langle S, A, \alpha' \rangle$ where $\alpha : S \rightarrow \mathcal{D}(S) + A \times S + 1$ and $\alpha' : S \rightarrow \mathcal{D}(S) + \mathcal{P}(A \times S)$ and

$$\alpha'(s) = \begin{cases} \{\alpha(s)\} & \text{if } \alpha(s) \in A \times S \\ \emptyset & \text{if } \alpha(s) = * \\ \alpha(s) & \text{otherwise} \end{cases}$$

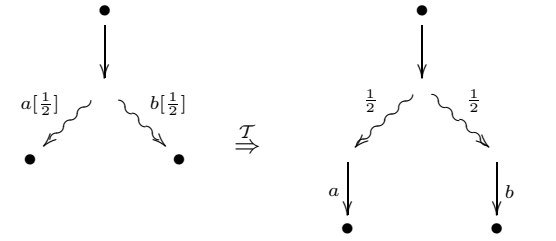
The diagram of a stratified automaton when translated to alternating automaton stays the same.

Seg $\rightarrow$ **PZ**: Let $\mathcal{A} = \langle S, A, \alpha \rangle$ be a Segala automaton, $\alpha : S \rightarrow \mathcal{P}(\mathcal{D}(A \times S))$. Then $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ where α' is determined from α in the following way:

$$\alpha'(s) = \{\mu' \mid \mu \in \alpha(s)\}$$

where μ' is constructed from μ by changing a distribution over pairs to distribution over singletons of pairs, i.e.

$$\mu' = \{\{\langle a, s' \rangle\} \mapsto \mu(a, s')\}.$$

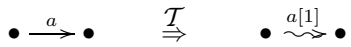


a Segala transition as a Pnueli-Zuck transition

Dirac arrows

Sometimes a transformation from an element to a Dirac distribution for this element is needed. This kind of translation embeds the non-probabilistic automata, **DLTS** and **LTS**, into the reactive and simple Segala automata, respectively.

DLTS $\rightarrow$ **React**: In this case every transition has to be changed to a probabilistic transition with probability 1.



a **DLTS** transition as a reactive transition

For $\langle S, A, \alpha \rangle \in$ **DLTS**, the translation is $\mathcal{T}(\langle S, A, \alpha \rangle) = \langle S, A, \alpha' \rangle$, if $\alpha(s) \in (S + 1)^A$, we put $\alpha'(s) \in (\mathcal{D}(S) + 1)^A$ such that $\alpha(s)(a) = t \in S \iff \alpha'(s)(a) = \mu_t^1$.

LTS $\rightarrow$ **SSeg**: For obtaining a simple Segala automaton out of an LTS, we change the next state of every transition to a Dirac distribution for this state.

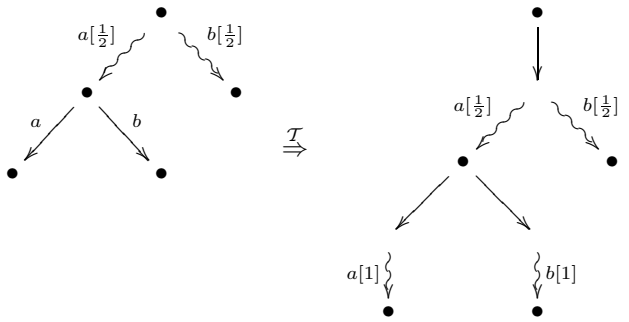
a **LTS** transition as a simple Segala transition

Formally, $\mathcal{T}(\langle S, A, \alpha \rangle) = \langle S, A, \alpha' \rangle$ such that $\alpha'(s) = \{\langle a, \mu_{s'}^1 \rangle \mid \langle a, s' \rangle \in \alpha(s)\}$.

Specific arrows

Var $\rightarrow$ **Seg**: Let $\mathcal{A} = \langle S, A, \alpha \rangle$ be a Vardi automaton, with $\alpha(s) : S \rightarrow \mathcal{D}(A \times S) \cup \mathcal{P}(A \times S)$. The translation to a Segala automaton is given by $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ for

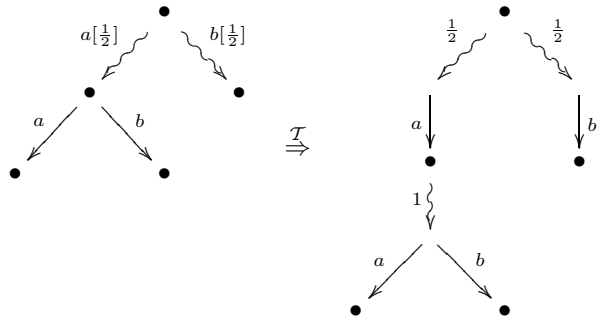
$$\alpha'(s) = \begin{cases} \{\alpha(s)\} & \text{if } \alpha(s) \in \mathcal{D}(A \times S) \\ \{\mu_{\langle a, s' \rangle}^1 \mid \langle a, s' \rangle \in \alpha(s)\} & \text{if } \alpha(s) \in \mathcal{P}(A \times S) \end{cases}$$



a Vardi system as a Segala system

Var → Bun: This translation is orthogonal to the one that gives us **Var → Seg**. For a Vardi automaton $\mathcal{A} = \langle S, A, \alpha \rangle$ we put $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ where

$$\alpha'(s) = \begin{cases} \{\{\langle a, s' \rangle\} \mapsto \mu(a, s')\} & \text{if } \alpha(s) = \mu \in \mathcal{D}(A \times S) \\ \mu^1_{\alpha(s)} & \text{if } \alpha(s) \in \mathcal{P}(A \times S) \end{cases}$$



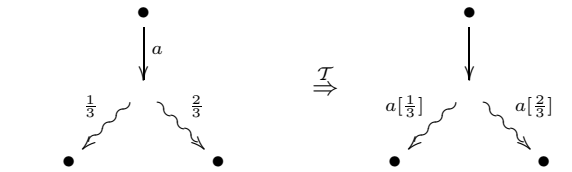
a Vardi system as a bundle system

Remark 2.4.2. Both in **Var → Seg** and in **Var → Bun** the translated transition function α' is well defined when we consider $\mathcal{D}(A \times S) \cap \mathcal{P}(A \times S) \neq \emptyset$, i.e. we identify $\mu^1_{\langle a,s \rangle}$ with $\{\langle a, s \rangle\}$. Furthermore, this identification is needed to obtain injectivity of the translations.

Alt → MG: Similarly as when translating Vardi systems, with an extra singleton construction, an alternating automaton $\mathcal{A} = \langle S, A, \alpha \rangle$, $\alpha : S \to \mathcal{D}(S) + \mathcal{P}(A \times S)$ is translated into a general probabilistic automaton. We put $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ where

$$\alpha'(s) = \begin{cases} \{\{\{s'\} \mapsto \mu(s')\}\} & \text{if } \alpha(s) = \mu \in \mathcal{D}(S) \\ \{\mu^1_{\alpha(s)}\} & \text{if } \alpha(s) \in \mathcal{P}(A \times S) \end{cases}$$

SSeg → Seg: In order to change a transition of a simple Segala automaton to a transition of a Segala automaton it is enough to push the action label into the distribution.



a simple Segala transition as a Segala transition

Formally, if $\mathcal{A} = \langle S, A, \alpha \rangle$, $\alpha : S \to \mathcal{P}(A \times \mathcal{D}(S))$ then $\mathcal{T}(\mathcal{A}) = \langle S, A, \alpha' \rangle$ where

$$\langle a, \mu \rangle \in \alpha(s) \iff \mu_a \in \alpha'(s) \in \mathcal{P}(\mathcal{D}(A \times S))$$

and

$$\mu_a(b, s') = \begin{cases} \mu(s') & \text{if } a = b \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

For this last translation, as an illustration, we give the proof of preservation and reflection of bisimilarity. Let μ be a distribution on S and a an action in A. Denote by μ_a the distribution on $A \times S$ obtained from μ by Equation (2.3), $\mu_a(a, s) = \mu(s)$. Clearly, for any subset $X \subseteq S$ we have $\mu[X] = \mu_a[a, X]$, which yields

$$\mu \equiv_R \mu' \iff \mu_a \equiv_{R,A} \mu'_a. \tag{2.4}$$

By the translation, the following holds:

1. If $s \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \mu$, then $s \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \mu_a$.
2. If $s \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \mu$ then there exists $a \in A$ such that $\mu = \nu_a$ for some distribution ν on S and $s \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \nu$.

Now assume $s \sim_{\mathcal{A}} t$, i.e. there exists a bisimulation R (Definition 2.2.3) with $\langle s, t \rangle \in R$. We prove that R is a bisimulation (Definition 2.2.4) for $\mathcal{T}(\mathcal{A})$. Let $s \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \mu$. By 2. we have that $\mu = \nu_a$ for some $a \in A$ and some $\nu \in \mathcal{D}(S)$, and $s \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \nu$. Since R is a bisimulation for $\mathcal{A}$, we have that there exists a distribution ν' such that $t \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \nu'$ and $\nu \equiv_R \nu'$. Now it follows by 1. that $s \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \nu'_a$, and furthermore $\mu = \nu_a \equiv_{A,R} \nu'_a$ by Equation (2.4). So, R is a bisimulation for $\mathcal{T}(\mathcal{A})$.

The opposite is analogous. If R is a bisimulation (Definition 2.2.4) with $\langle s, t \rangle \in R$ for $\mathcal{T}(\mathcal{A})$, we prove that R is a bisimulation (Definition 2.2.3) for $\mathcal{A}$. Assume $s \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \mu$. Then by 1., $s \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \mu_a$. Since R is a bisimulation and $\langle s, t \rangle \in R$ we get that there exists $\nu \in \mathcal{D}(A \times S)$ such that $t \rightarrow\!\!\rightsquigarrow_{\mathcal{T}(\mathcal{A})} \nu$ and $\mu_a \equiv_{R,A} \nu$. Now, by 2., we get that there exists $a' \in A$ and a distribution μ' on S such that $\nu = \mu'_{a'}$, $t \xrightarrow{a'}\!\!\rightsquigarrow_{\mathcal{A}} \mu'$ and $\mu \equiv_R \mu'$. However, since $\mu_a \equiv_{A,R} \nu = \mu'_{a'}$ by Equation (2.3) and Equation (2.4) it follows that $a' = a$ and hence $t \xrightarrow{a}\!\!\rightsquigarrow_{\mathcal{A}} \mu'$, which completes the proof.

Strict alternation vs. complex transition function

A translation between simple Segala automata and automata of the class $\mathbf{SA}_n$ has been known for a long time, and has recently been justified in [BS01]. Similarly, the class $\mathbf{SA}_p$ can be compared with the class of bundle probabilistic automata. In order to carry out these comparisons in our framework we slightly change the comparison criterion itself so that it allows for translations that do not keep the same state set.

Relaxed expressiveness criteria

Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be two classes of probabilistic automata and let $\mathcal{T} : \mathbf{C}_1 \to \mathbf{C}_2$ be a translation mapping, such that if $\mathcal{A} = \langle S, A, \alpha \rangle$ then $\mathcal{T}(\mathcal{A}) = \langle S', A, \alpha' \rangle$

We say that the class $\mathbf{C}_1$ is *relaxed embedded* in the class $\mathbf{C}_2$, by the translation $\mathcal{T}$, notation $\mathbf{C}_1 \twoheadrightarrow \mathbf{C}_2$ if the following conditions hold:

1. for any automaton of $\mathbf{C}_1$, $S \subseteq S'$ or for any automaton of $\mathbf{C}_1$, $S' \subseteq S$
2. bisimilarity is preserved and reflected for common states, i.e., $\forall s, t \in S \cap S' \colon s \sim_{\mathcal{A}} t \iff s \sim_{\mathcal{T}(\mathcal{A})} t$, and
3. the translation $\mathcal{T}$ is injective.

Furthermore, we say that the class $\mathbf{C}_1$ *is embedded up to irrelevant bisimilarity* in the class $\mathbf{C}_2$, by the translation $\mathcal{T}$, notation $\mathbf{C}_1 \longrightarrow_{/\sim} \mathbf{C}_2$ if the following conditions hold:

1. $S' \subseteq S$
2. for any two common states $s, t \in S'$, $s \sim_{\mathcal{A}} t \iff s \sim_{\mathcal{T}(\mathcal{A})} t$, and
3. the translation $\mathcal{T}$ is injective up to bisimilarity of irrelevant states, i.e., if $\mathcal{A}_1 = \langle S_1, A, \alpha_1 \rangle$ and $\mathcal{A}_2 = \langle S_2, A, \alpha_2 \rangle$ are two automata of the first class such that $\mathcal{T}(\mathcal{A}_1) = \mathcal{T}(\mathcal{A}_2) = \langle S, A, \alpha \rangle$, then
$$s_1 \in S_1 \setminus S \Rightarrow \exists s_2 \in S_2 \setminus S \colon s_1 \sim s_2 \text{ and}$$
$$s_2 \in S_2 \setminus S \Rightarrow \exists s_1 \in S_1 \setminus S \colon s_1 \sim s_2.$$

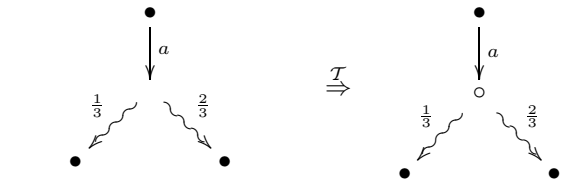
The second relaxed expressiveness criterion relaxes the injectivity assumption. It might be that different systems are translated to the same one, but the difference is only due to presence of some bisimilar states. It allows us to show that the class $\mathbf{SA}_n$ can be embedded in the class **SSeg** which is intuitively expected, a translation is given by removing the non-deterministic states.

Theorem 2.4.3. *The embeddings from Figure 2.2 explain the relationships between the classes of strictly alternating, bundle and simple Segala probabilistic automata.*

Figure 2.2: Strictly alternating models as models with a structured transition relation

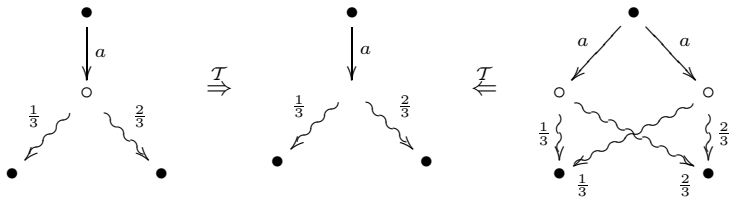
We give the translations needed for each of the embeddings.

SSeg $\twoheadrightarrow$ **SA**$_n$:

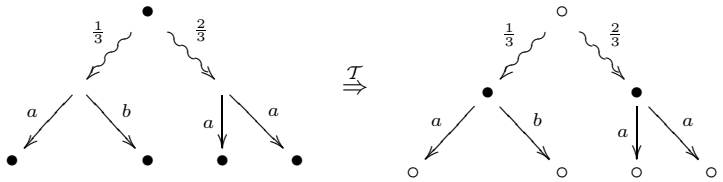


Let $\mathcal{A} = \langle S, A, \alpha \rangle$ be a simple Segala automaton. The translation function translates it to $\mathcal{T}(\mathcal{A}) = \langle S', A, \alpha' \rangle$ by inserting a new probabilistic state for each transition, i.e., by changing $s \xrightarrow{a}\!\!\rightsquigarrow \mu$ to $s \xrightarrow{a} s_{s,a,\mu} \rightsquigarrow \mu$. Formally, $S' = S + S_p$ where S_p is a set of fresh probabilistic states, such that $S_p = \{s_{s,a,\mu} \mid s \in S, a \in A, \mu \in \mathcal{D}(S), \langle a, \mu \rangle \in \alpha(s)\}$. So, for every outgoing transition of a state there is a corresponding new state added to S_p. The transition function is defined as follows: if $s \in S$, such that $\langle a, \mu \rangle \in \alpha(s)$ and the corresponding new state is $s_{s,a,\mu}$ then and only then $\langle a, s_{s,a,\mu} \rangle \in \alpha'(s)$; if $s_{s,a,\mu} \in S_p$ corresponds to a transition $\langle a, \mu \rangle \in \alpha(s)$ then $\alpha'(s_{s,a,\mu}) = \mu$. The translation is injective and it preserves and reflects bisimilarity on S.

SA$_n$ $\longrightarrow\!\!\!\!\!\rightarrow_{/\sim}$ **SSeg**: The translation for this embedding, is the "inverse" of the case **SSeg** $\twoheadrightarrow$ **SA**$_n$. If $\mathcal{A} = \langle S, A, \alpha \rangle \in$ **SA**$_n$ with a set of probabilistic states P and a set of non-deterministic states N, then $\mathcal{T}(\mathcal{A}) = \langle N, A, \alpha' \rangle$ where for every $s \in N$, $\alpha'(s) = \{\langle a, \mu \rangle \mid \langle a, s' \rangle \in \alpha(s)$ and $\alpha(s') = \mu\}$. Hence this translation forgets the probabilistic states, and turns two strictly alternating steps of the **SA**$_n$ automaton into one transition of the corresponding Segala automaton. This translation is only injective up to bisimilarity of the disappearing probabilistic states. In fact it is even stricter than that, it is injective up to indistinguishable probabilistic states. The only cases of non-injectivity come as a consequence of having two probabilistic states s_{p1} and s_{p2} such that $\alpha(s_{p1}) = \alpha(s_{p2}) = \mu$. Then if s is a non-deterministic state with $s \xrightarrow{a} s_{p1}$ and $s \xrightarrow{a} s_{p2}$, the two "copies" $s \xrightarrow{a} s_{p1} \rightsquigarrow \mu$ and $s \xrightarrow{a} s_{p2} \rightsquigarrow \mu$ will be mapped to one single transition $s \xrightarrow{a}\!\!\rightsquigarrow \mu$. For example, the following two different **SA**$_n$ automata translate to one **SSeg** automaton.



Bun $\twoheadrightarrow$ $\mathbf{SA}_p$:



a bundle automaton as a $\mathbf{SA}_p$ automaton

The translation is as follows: $\mathcal{T}(\langle S, A, \alpha\rangle) = \langle S', A, \alpha'\rangle$ where $S' = S + S_n$, S_n containing the fresh non-deterministic states $S_n = \{s_{s,T} \mid s \in S, T \in \mathrm{supp}(\alpha(s))\}$. The transition function is defined by: for $s \in S$ with $\alpha(s) = \mu$, we put $\alpha'(s) = \mu_s \in \mathcal{D}(S_n)$ where $\mu_s(s_{s,T}) = \mu(T)$, and for $s_{s,T} \in S_n$ we put $\alpha'(s_{s,T}) = T$.

$\mathbf{SA}_p \twoheadrightarrow$ **Bun**: The inverse of the translation **Bun** $\twoheadrightarrow \mathbf{SA}_p$ gives us the translation in this case. We forget all the non-deterministic states in a $\mathbf{SA}_p$ automaton, being left with a bundle automaton. For a $\mathbf{SA}_p$ automaton $\mathcal{A} = \langle S, A, \alpha\rangle$ with a set of probabilistic states P and non-deterministic states N we put $\mathcal{T}(\mathcal{A}) = \langle P, A, \alpha'\rangle$ where for $s \in P$ with $\alpha(s) = \mu \in \mathcal{D}(N)$ we define $\alpha'(s) = \mu_s \in \mathcal{D}(\mathcal{P}(A \times S))$ such that $\mu_s(\alpha(s')) = \mu(s')$. Note that the phenomenon of losing "copies" as in $\mathbf{SA}_n \longrightarrow\!\!\!\!/_{\sim}$ **SSeg** does not appear here. Even though states are lost, no transitions are identified, i.e. no arrow is lost.

Remark 2.4.4. Restricting to subclasses of the class **SA**, i.e. considering the classes $\mathbf{SA}_n$ and $\mathbf{SA}_p$, is necessary to obtain injectivity (up to bisimilar irrelevant states) of the embedding mappings.

We note that recently a very detailed and interesting comparative analysis of bisimulation relations on alternating and non-alternating (Segala) models has been carried out by Segala and Turrini [ST05].

2.5 Summary of the chapter

In this chapter we have presented various types of probabilistic automata, including generative, reactive and stratified ones, strictly alternating and alternating ones, the simple Segala, Segala and Vardi type of probabilistic automata and the bundle, Pnueli-Zuck and general ones.

A major part of our work has been devoted to the comparison of the various classes of probabilistic automata, taking strong bisimilarity for these automata as a starting point, resulting in a hierarchy of probabilistic system types. In addition, we have discussed to which extent non-determinism can be modelled in the various types of automata and the operator of parallel composition for them. Classes positioned higher in the map of Figure 2.1 can be characterized as closures of the simpler classes under parallel composition, which clarifies the need for the more complex models.

The results obtained are briefly presented in Figure 2.1, Figure 2.2 and Table 2.1. From there various conclusions on probabilistic system modelling can be drawn. For instance, if presence of non-determinism and being closed under all variants of parallel composition are desired properties on the one hand, and having as simple a model as possible is needed on the other hand, then whether the choice is for input (reactive) or output (generative) type of systems, the best choice appears to be the simple Segala model and the bundle model, respectively. Different requirements lead to different choices, but we hope the map of probabilistic automata based models will prove to be useful in making a right modelling decision.

3

Categories and coalgebras

In this chapter we present some notions and results from category theory and the theory of coalgebras that will be used in the sequel. In the first two sections we present standard notions and fix the notation. This sets the preliminaries for the chapters that follow. The last section is devoted to the characterization of coalgebraic bisimulation in terms of concrete transfer conditions.

In the sequel we will adopt an abstract point of view in which we treat the probabilistic systems and their equivalences. An abstract theory that provides a uniform framework for various kinds of transition systems is the theory of coalgebras [Rut96, JR96, Rut00, Gum99], arising as a rather recent theory within, or closely connected to category theory [Mac71, Bor94]. In this chapter we introduce the basics of category theory and coalgebras, and set up the notation for the sequel. Furthermore, we also present a couple of new results of coalgebraic nature that are to be used in the remaining chapters and are collected here.

The chapter is organized as follows: In Section 3.1 we provide basic definitions from category theory, such as category and functor. Section 3.2 introduces the central notion of this thesis, coalgebras, and presents basic examples. There we also define the notion of action-type coalgebras. We proceed by presenting more basic notions from category theory, namely natural transformations, limits and colimits in Section 3.3. In the study of dynamic systems one is often interested in the states of the system up to some kind of behavioral equivalence: states that exhibit the same behavior can be considered "the same". One of these behavioral equivalences is bisimilarity, arising from the notion of a bisimulation relation. Bisimulations are treated in Section 3.4. Section 3.5 focuses on more examples of coalgebras: we view transition systems as coalgebras and we specify the functors whose coalgebras we will focus on. As a new technical result we note the proof of the weak pullback preservation for the probabilistic functor. In Section 3.6, we present a method for characterizing coalgebraic bisimilarity in terms of transfer conditions. We provide transfer conditions for the basic functors and we show how transfer conditions can be derived for composed functors, in an inductive way.

3.1 Basic notions of category theory

In this section we mention the most basic notions from category theory. For more detail, the interested reader is referred to, for example, the textbooks by MacLane and Borceux [Mac71, Bor94], respectively. In many cases we adopt the formulation of the definitions from [Gum99].

Categories

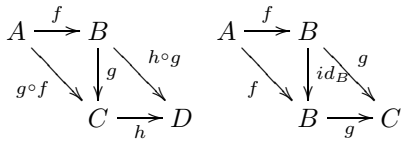
A *category* $\mathbf{C}$ consists of a class of objects, $\mathbf{C}_o$ and a class of arrows or morphisms, $\mathbf{C}_m$ between the objects together with the following operations:

- $dom : \mathbf{C}_m \to \mathbf{C}_o$
- $codom : \mathbf{C}_m \to \mathbf{C}_o$
- $id : \mathbf{C}_o \to \mathbf{C}_m$.

The operation *dom* associates to each arrow its domain (source object), *codom* associates to each arrow its codomain (target object). If f is an arrow with $dom(f) = A$ and $codom(f) = B$ one writes, $f : A \to B$. The operation *id* assigns to an object its identity arrow, usually denoted by $id_A : A \to A$ (or 1_A) for an object A. Moreover, there is a partial operation of composition of arrows. The composition of the arrows f and g is defined if and only if $codom(f) = dom(g)$, i.e. if $f : A \to B, g : B \to C$. The result is an arrow $g \circ f : A \to C$. The following laws have to be satisfied whenever the composition is defined

- $(h \circ g) \circ f = h \circ (g \circ f)$
- $id_B \circ f = f$ and $g \circ id_B = g$.

It is common to represent objects and arrows via diagrams. In a diagram the objects are drawn as points and the morphisms as arrows. Composition of arrows is often not explicitly drawn, but its presence is implied. A path of arrows represents the composition of the involved arrows. If there are two different paths from an object to another object that enclose an area, it is implied that the corresponding compositions are equal. One says that the diagram (or parts of it) commutes. The above laws that the composition of arrows has to satisfy can be expressed by the commutativity of the following two diagrams.



Example 3.1.1. Some examples of categories are:

1. The category Set whose objects are sets and arrows are functions.
2. The category $\mathsf{Set}\times\mathsf{Set}$ is a product category whose objects are pairs of sets, and arrows are pairs of functions $\langle f, g\rangle : \langle A, B\rangle \to \langle C, D\rangle$ for $f : A \to C$ and $g : B \to D$. The composition of arrows is done component wise. This way one defines also products of arbitrary categories.
3. We write Rel for the category of binary relations. Its objects are arbitrary relations $A \subseteq X \times Y$ and a morphism between $A_1 \subseteq X_1 \times Y_1$ and $A_2 \subseteq X_2 \times Y_2$ is a pair of functions $\langle f, g\rangle$ as above for $f : X_1 \to X_2$ and $g : Y_1 \to Y_2$ which preserve the relation, in the sense that $\langle x_1, y_1\rangle \in A_1 \implies \langle f(x_1), g(y_1)\rangle \in A_2$.

Throughout the thesis we will mainly deal with the category Set.

A morphism f in a category is called *mono* if it is left cancellable i.e. if for all g_1 and g_2, $f \circ g_1 = f \circ g_2 \implies g_1 = g_2$, and it is called *epi* if it is right cancellable, $g_1 \circ f = g_2 \circ f \implies g_1 = g_2$. In order to emphasize that $f : A \to B$ is mono, we will sometimes write $f : A \hookrightarrow B$. Similarly, for an epi morphism $f : A \to B$ we write $f : A \twoheadrightarrow B$. A morphism is an isomorphism if it is invertible. In Set monos are the injective functions, epis are the surjective functions and isomorphisms are the bijective functions.

Functors

A *functor* $\mathcal{F}$ between two categories $\mathbf{C}$ and $\mathbf{D}$ consists of two maps

- $\mathcal{F}_o : \mathbf{C}_o \to \mathbf{D}_o$, between the objects and
- $\mathcal{F}_m : \mathbf{C}_m \to \mathbf{D}_m$, between the morphisms

where for a morphism $f : A \to B$ from $\mathbf{C}_m$, $\mathcal{F}_m(f) : \mathcal{F}_o(A) \to \mathcal{F}_o(B)$ is a morphism in $\mathbf{D}_m$ such that composition and identity are respected, i.e.,

- $\mathcal{F}_m(g \circ f) = \mathcal{F}_m(g) \circ \mathcal{F}_m(f)$
- $\mathcal{F}_m(id_A) = id_{\mathcal{F}_o(A)}$.

Usually the indices $_m$ and $_o$ of a functor $\mathcal{F}$ are dropped. Moreover, often the brackets are also dropped and one writes $\mathcal{F}A$ and $\mathcal{F}f$ instead of $\mathcal{F}(A)$ and $\mathcal{F}(f)$, respectively. We will often consider *endofunctors*, functors from a category to itself.

Example 3.1.2. The following are examples of Set endofunctors.

1. The identity functor $\mathcal{I}d$ maps every set and every function to itself.

2. Let A be a fixed set. The constant functor $\underline{A}$ maps every set to A and every function to the identity on A, id_A.

3. The powerset functor $\mathcal{P}$ maps any set to the set of its subsets,

$$\mathcal{P}X = \{Z \mid Z \subseteq X\}$$

and on functions $f : X \to Y$ it is given by

$$(\mathcal{P}f)Z = f(Z) = \{f(z) \mid z \in Z\}.$$

4. Let A be a fixed set. The constant exponent functor $\mathcal{I}d^A$ maps any set to the set of all functions from A to itself, i.e.

$$\mathcal{I}d^A(X) = X^A = \{f : A \to X\},$$

and it maps a function $f : X \to Y$ to the function $\mathcal{I}d^A(f) : X^A \to Y^A$ defined by

$$\mathcal{I}d^A(f)(\xi) = f \circ \xi \text{ for } \xi : A \to X.$$

It is important to note that all Set endofunctors preserve epis and also preserve monos with non-empty domain. That is, if f is epi or mono with non-empty domain, then $\mathcal{F}f$ is also epi or mono, respectively. Namely, let $f : A \twoheadrightarrow B$ be an epi in Set and let $\mathcal{F}$ be a Set endofunctor. Then f is a surjective function and therefore it has a right inverse $f' : B \to A$ such that $f \circ f' = id_B$. From the functorial properties of $\mathcal{F}$ we have that $\mathcal{F}f \circ \mathcal{F}f' = \mathcal{F}(f \circ f') = \mathcal{F}(id_B) = id_{\mathcal{F}B}$ i.e. $\mathcal{F}f$ also has a right inverse and is therefore a surjective function, i.e. an epi in Set. Similarly, every injective function with non-empty domain has a left inverse and is therefore preserved by a Set endofunctor.

A *bifunctor* on Set is any functor from $\mathsf{Set} \times \mathsf{Set}$ to Set[1]. If $\mathcal{F}$ is a bifunctor, and A is a fixed set, then a Set endofunctor $\mathcal{F}_A$ is defined by

$$\mathcal{F}_A S = \mathcal{F}\langle A, S\rangle, \quad \mathcal{F}_A f = \mathcal{F}\langle id_A, f\rangle, f : S \to T. \tag{3.1}$$

Another way to get a Set endofunctor from a bifunctor is by applying it to two copies of the same set. Namely, let $\mathcal{F}$ be a bifunctor. Then $\mathcal{F}_\Delta$ is a Set endofunctor, by putting

$$\mathcal{F}_\Delta S = \mathcal{F}\langle S, S\rangle, \quad \mathcal{F}_\Delta f = \mathcal{F}\langle f, f\rangle, f : S \to T. \tag{3.2}$$

Note that any set endofunctor can be considered as a bifunctor that does not depend on one of its arguments. Namely, for a Set endofunctor $\mathcal{F}$, a corresponding bifunctor $\hat{\mathcal{F}}$ can be defined by

$$\hat{\mathcal{F}}\langle S, T\rangle = \mathcal{F}T, \quad \hat{\mathcal{F}}\langle f, g\rangle = \mathcal{F}g, f : S \to S', g : T \to T'. \tag{3.3}$$

[1] In general, a bifunctor is a functor from a product category into a category, but we restrict our attention to bifunctors on Set only.

Then clearly,

$$\hat{\mathcal{F}}_A = \mathcal{F}. \tag{3.4}$$

An example of a bifunctor is the product $\times : \mathsf{Set} \times \mathsf{Set} \to \mathsf{Set}$. It maps sets A and B to their Cartesian product $A \times B$. A pair of functions $f : A \to A'$, $g : B \to B'$ is mapped to $f \times g : A \times B \to A' \times B'$ where $(f \times g)(\langle a, b\rangle) = \langle f(a), g(b)\rangle$.

3.2 Coalgebras

In this section we define the central notion of the thesis, coalgebra of a functor, and we present several examples. Coalgebras of a functor are structures dual to algebras of a signature. They are a perfect abstract tool for modelling dynamic systems and infinite data structures. Moreover, once a system is identified as a coalgebra of a functor, many notions and results from the theory of universal coalgebra can be used for its analysis. Such notion is for example the notion of a bisimulation relation between coalgebras.

The theory of universal coalgebras was systematically treated for the first time by Rutten [Rut96, Rut00]. Further introductory texts on the subject can be found in the articles by Jacobs and Rutten [JR96], Jacobs [Jac02] and Gumm [Gum99].

Consider a signature $\Sigma = \{*\}$ consisting of a single operation symbol $*$ which is binary. A concrete algebra of the signature Σ is a groupoid, i.e. a pair $\langle A, *_A\rangle$ consisting of a set A and a binary operation $*_A : A \times A \to A$. Since the signature only determines the arity of the operation, we can say that an algebra of signature (type) Σ is a pair $\langle A, \alpha : \times_\Delta A \to A\rangle$. Here $\times_\Delta$ is the functor obtained from the bifunctor $\times$ as in Equation (3.2). In general, an algebra of type $\mathcal{F}$ is a pair $\langle A, \alpha\rangle$ for $\alpha : \mathcal{F}A \to A$. Dually, the notion of coalgebra of a type, i.e. coalgebra for a functor, arises.

Definition 3.2.1. *Let $\mathcal{F}$ be a* Set*-functor. A coalgebra for $\mathcal{F}$, or an* $\mathcal{F}$-coalgebra*, for short, is a pair $\langle S, \alpha\rangle$ where S is a carrier set, or a set of states, and $\alpha : S \to \mathcal{F}S$ is a transition function, a structure map.*

A homomorphism *between two $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ is a function $h : S \to T$ satisfying $\mathcal{F}h \circ \alpha = \beta \circ h$, i.e. making the following diagram commute.*

$$\begin{array}{ccc} S & \xrightarrow{h} & T \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ \mathcal{F}S & \xrightarrow{\mathcal{F}h} & \mathcal{F}T \end{array}$$

The $\mathcal{F}$-coalgebras together with their homomorphisms form a category, which we denote by $\mathsf{Coalg}_{\mathcal{F}}$.

Example 3.2.2. Consider the transition systems of Definition 2.1.8. They are coalgebras of the powerset functor, i.e. pairs $\langle S, \alpha : S \to \mathcal{P}S\rangle$. If we instantiate the definition of a coalgebra homomorphisms for the particular case of the functor $\mathcal{P}$, we easily get that a function $h : S \to T$ is a homomorphism between the transition systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if for any state $s \in S$, whenever $s \to s'$ then $h(s) \to h(s')$, and the other way around, i.e., if $h(s) \to t$ then there exists a state $s' \in S$ such that $h(s') = t$ and $s \to s'$.

Example 3.2.3. Let A be a fixed set (of actions). Let $\mathcal{B}$ be defined by

$$\mathcal{B}S = \mathcal{P}(A \times S)$$

and for $f : S \to T, V \subseteq A \times S$,

$$\mathcal{B}f(V) = \{\langle a, f(s)\rangle \mid \langle a, s\rangle \in V\}.$$

Then $\mathcal{B}$ is a functor and the $\mathcal{B}$-coalgebras are the labelled transition systems (Definition 2.1.9). A function $h : S \to T$ is a homomorphism between the LTSs $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if for any state $s \in S$, whenever $s \xrightarrow{a} s'$ then $h(s) \xrightarrow{a} h(s')$, and if $h(s) \xrightarrow{a} t$ then there exists a state $s' \in S$ such that $h(s') = t$ and $s \xrightarrow{a} s'$.

In the definition of LTSs and in other cases the action set A plays an important role. When we wish to emphasize the role of the actions, as will be the case in Chapter 5, we speak of action-type coalgebras.

Definition 3.2.4. *An action-type coalgebra, with action set A, is a triple $\langle S, A, \alpha\rangle$ such that $\langle S, \alpha\rangle$ is an $\mathcal{F}_A$-coalgebra, where $\mathcal{F}_A$ is the functor induced by a bifunctor $\mathcal{F}$, as in Equation (3.1).*

Example 3.2.5. Consider the product bifunctor $\times$ defined in the previous section. Let A be a fixed set of actions. A triple $\langle S, A, \alpha\rangle$ where $\alpha : S \to A \times S$ is an action-type coalgebra of the functor $\times_A$. The $\times_A$-coalgebras are the fully deterministic systems, in which from any state a single transition with label from A leads to the next state.

Note that Equation (3.4), together with Equations (3.3) and (3.1), shows that "action-type" coalgebras only emphasize the action set and do not restrict the class of all coalgebras, i.e., any coalgebra can trivially be considered an action-type coalgebra.

We will consider coalgebraic bisimulations and present more examples of coalgebras in Section 3.4 and Section 3.5 below. First, we need to introduce some more notions from category theory.

3.3 More basic notions from category theory

In this section we focus on natural transformations, limits and colimits.

Natural transformations

Let $\mathcal{F}$ and $\mathcal{G}$ be functors from a category $\mathbf{C}$ to a category $\mathbf{D}$. A *natural transformation* τ from $\mathcal{F}$ to $\mathcal{G}$, notation $\tau : \mathcal{F} \Longrightarrow \mathcal{G}$, is a family of arrows $\tau = \{\tau_A : \mathcal{F}A \to \mathcal{G}A\}$ indexed by objects A of $\mathbf{C}$, such that for any arrow $f : A \to B$ in $\mathbf{C}$, the following naturality diagram commutes.

$$\begin{array}{ccc} \mathcal{F}A & \xrightarrow{\mathcal{F}f} & \mathcal{F}B \\ {\scriptstyle \tau_A}\downarrow & & \downarrow{\scriptstyle \tau_B} \\ \mathcal{G}A & \xrightarrow{\mathcal{G}f} & \mathcal{G}B \end{array}$$

Hence natural transformations transform the structure constructed by $\mathcal{F}$ to the structure constructed by $\mathcal{G}$. The requirement that the above diagram commutes, the "naturality condition", expresses that the transformation is defined in a uniform way that works the same for any object of the category $\mathbf{C}$.

Let $\mathbf{C}$ and $\mathbf{D}$ be categories. Then the collection of functors from $\mathbf{C}$ to $\mathbf{D}$ form a category with morphisms natural transformations.

Example 3.3.1. Examples of natural transformations are the following.

1. Let $\mathcal{F}$ be any functor from a category $\mathbf{C}$. The identity natural transformation $id : \mathcal{F} \Longrightarrow \mathcal{F}$ is the family of identity arrows, i.e. $id = \{i_X = id_{\mathcal{F}X}\}$ indexed by objects of $\mathbf{C}$. It is obvious that the naturality condition is satisfied in this case.

2. Let $\sigma_X : X \to \mathcal{P}X$ be given by $\sigma_X(x) = \{x\}$. Then $\sigma = \{\sigma_X\}$ indexed by sets is a natural transformation from the Set endofunctor $\mathcal{I}d$ to the Set endofunctor $\mathcal{P}$, $\sigma : \mathcal{I}d \Longrightarrow \mathcal{P}$. Namely, for any element $x \in \mathcal{F}A$ we have $\sigma_B \circ \mathcal{I}df(x) = \{f(x)\}$ and $\mathcal{P}f \circ \sigma_A(x) = \mathcal{P}f(\{x\}) = f(\{x\}) = \{f(x)\}$, showing that the naturality condition is satisfied.

3. Let A be a fixed set. For any set X, let $\eta_X : \underline{A}X \to \mathcal{P}X$ denote the function defined by $\eta_X(a) = \emptyset$ for all $a \in A$. Then one easily verifies that $\eta = \{\eta_X\}$ is a natural transformation, $\eta : \underline{A} \Longrightarrow \mathcal{P}$.

We formulate the next simple lemma taken from [Bor94] for further reference.

Lemma 3.3.2. *Let $\mathcal{F}$ be a bifunctor. Let A_1 and A_2 be two fixed sets, and $f : A_1 \to A_2$ a mapping. Then f induces a natural transformation $\tau^f : \mathcal{F}_{A_1} \Longrightarrow \mathcal{F}_{A_2}$ defined by $\tau^f_S = \mathcal{F}\langle f, id_S\rangle$.* □

To compare the expressiveness of coalgebras for different functors, say $\mathcal{F}$ and $\mathcal{G}$, we will study translations of $\mathcal{F}$-coalgebras into $\mathcal{G}$-coalgebras. Such a translation can easily be obtained from a natural transformation between the two functors under consideration (cf. [Rut00]).

Definition 3.3.3. *A natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ gives rise to a functor $\mathcal{T}_\tau : \mathsf{Coalg}_\mathcal{F} \to \mathsf{Coalg}_\mathcal{G}$ defined for an $\mathcal{F}$-coalgebra $\langle S, \alpha\rangle$ and an $\mathcal{F}$-homomorphism h as*

$$\mathcal{T}_\tau\langle S, \alpha\rangle = \langle S, \tau_S \circ \alpha\rangle \quad \textit{and} \quad \mathcal{T}_\tau h = h.$$

To see that the above definition really defines a functor, we need to check that a homomorphism h between two $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ is also a homomorphism between the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle S, \alpha\rangle$ and $\mathcal{T}_\tau\langle T, \beta\rangle$. This follows easily from the naturality of τ:

$$\begin{array}{ccc} S & \xrightarrow{\quad h \quad} & T \\ {\scriptstyle\alpha}\downarrow & \text{assumption } h & \downarrow{\scriptstyle\beta} \\ \mathcal{F}S & \xrightarrow{\quad \mathcal{F}h \quad} & \mathcal{F}T \\ {\scriptstyle\tau_S}\downarrow & \text{naturality } \tau & \downarrow{\scriptstyle\tau_T} \\ \mathcal{G}S & \xrightarrow[\quad \mathcal{G}h \quad]{} & \mathcal{G}T \end{array}$$

Limits, colimits and their preservation

Let D be a diagram, that is, a collection of objects $(D_i)_{i\in I}$ and a collection of arrows $(f_k)_{k\in K}$ between the objects in $(D_i)_{i\in I}$.

A *cone* over D is a single object S together with morphisms $s_i : S \to D_i$ for each $i \in I$, such that for every arrow of the diagram $f_k : D_i \to D_j$ we have $f_k \circ s_i = s_j$.

A cone $\langle S, (s_i)_{i\in I}\rangle$ is a *weak limit* of D if for any other cone over D, $\langle S', (s'_i)_{i\in I}\rangle$, there is a morphism $s : S' \to S$ (also called mediating morphism) such that $s'_i = s_i \circ s$ for all $i \in I$.

A cone $\langle S, (s_i)_{i\in I}\rangle$ is the *limit* of D if it is a weak limit and the mediating morphism s is unique. The s_i are then called canonical morphisms.

Colimits are dual notions to limits. More precisely, a *co-cone* over D, $\langle C, (c_i)_{i\in I}\rangle$, is an object C together with a collection of arrows $c_i : D_i \to C$ such that for every arrow of the diagram $f_k : D_i \to D_j$ we have $c_j \circ f_k = c_i$. A co-cone is a *weak colimit* if for every competing co-cone $\langle C', (c'_i)_{i\in I}\rangle$ over D, there exists a mediating morphism $c : C \to C'$ with $c \circ c_i = c'_i$. A co-cone is the *colimit* of D if the mediating morphism is unique.

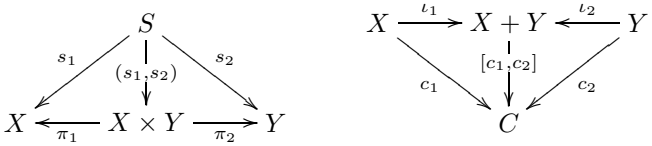
Limits and colimits, if they exist, are unique up to isomorphism. In Set all limits and colimits exist. Since limits over specific kinds of diagrams will play an important role, we will discuss them in the remainder of the section.

A *span* and a *cospan* between two objects X and Y are triples $\langle S, s_1, s_2\rangle$ and $\langle C, c_1, c_2\rangle$ of objects S and C and arrows as pictured below.

$$X \xleftarrow{s_1} S \xrightarrow{s_2} Y \qquad\qquad X \xrightarrow{c_1} C \xleftarrow{c_2} Y$$

Hence, on the one hand a span (cospan) is a cone (co-cone) over a diagram with two objects X and Y, and no arrows, and, on the other hand it is a diagram with three objects (X, Y and S or C, respectively) and two arrows between them.

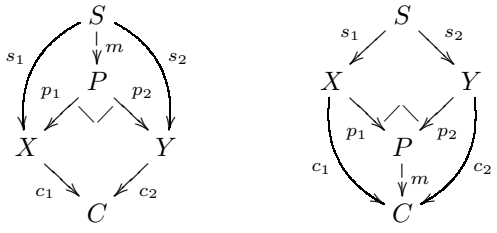
The limit and the colimit of a diagram with two objects and no arrows are called the *product* and *coproduct*, respectively. By $X \times Y$, with projections as canonical morphisms $\pi_1 : X \times Y \to X$ and $\pi_2 : X \times Y \to Y$ we denote the categorical product of two objects X and Y. By $X + Y$, with injections $\iota_1 : X \to X + Y$ and $\iota_2 : Y \to X + Y$ we denote the categorical coproduct of the two objects X and Y. This means that, for any span $\langle S, s_1, s_2\rangle$ and cospan $\langle C, c_1, c_2\rangle$ between X and Y, there exist unique functions $(s_1, s_2)\colon S \to X \times Y$ and $[c_1, c_2]\colon X + Y \to C$ making both parts of the respective diagram below commute.



The categorical products and coproducts in Set are simply Cartesian products, with projections, and disjoint unions, with injections. More precisely, let X and Y be two sets. The product is then $X \times Y = \{\langle x, y\rangle \mid x \in X, y \in Y\}$ and the projections are defined as usual, by $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$. The coproduct is $X + Y = \{\langle x, 1\rangle \mid x \in X\} \cup \{\langle y, 2\rangle \mid y \in Y\}$ and the injections are defined by $\iota_1(x) = \langle x, 1\rangle$ and $\iota_2(y) = \langle y, 2\rangle$.

We say that a span in Set, $\langle S, s_1, s_2\rangle$, between sets X and Y is *jointly injective* if $(s_1, s_2) : S \to X \times Y$ is injective. Dually, a cospan $\langle C, c_1, c_2\rangle$ is *jointly surjective* if $[c_1, c_2] : X + Y \to C$ is surjective. A relation $R \subseteq X \times Y$ gives rise to the jointly injective span $\langle R, \pi_1, \pi_2\rangle$ between X and Y.

Next we view spans and cospans as diagrams. The limit of a cospan $\langle C, c_1, c_2\rangle$ is called the *pullback*. Hence, a pullback is a span $\langle P, p_1, p_2\rangle$ as in the left diagram below satisfying $c_1 \circ p_1 = c_2 \circ p_2$ and such that for every span $\langle S, s_1, s_2\rangle$ with $c_1 \circ s_1 = c_2 \circ s_2$ there exists a unique mediating arrow $m : S \to P$ satisfying $s_1 = p_1 \circ m$ and $s_2 = p_2 \circ m$. The colimit of a span $\langle S, s_1, s_2\rangle$ is called the *pushout*. Hence, the pushout is a cospan $\langle P, p_1, p_2\rangle$ as in the right diagram below, such that for every cospan $\langle C, c_1, c_2\rangle$ with $c_1 \circ s_1 = c_2 \circ s_2$ there exists a unique mediating arrow $m : P \to C$ satisfying $c_1 = m \circ p_1$ and $c_2 = m \circ p_2$.



Extra wedges, as in the above diagrams, are used to emphasize that a certain diagram is a pullback or a pushout diagram, respectively.

According to the general definition, a *weak pullback* is a pullback for which the mediating arrow m need not be unique.

A pullback of a cospan $\langle C, c_1, c_2\rangle$ between sets X and Y is the span arising from the relation

$$Q := \{\langle x, y\rangle \in X \times Y \mid c_1(x) = c_2(y)\}.$$

A pushout of a span $\langle S, s_1, s_2\rangle$ between sets X and Y is the cospan $\langle P, p_X, p_Y\rangle$ with $P = (X+Y)/\theta$ for θ the smallest equivalence relation on $X+Y$ containing the pairs $\langle \iota_1(s_1(s)), \iota_2(s_2(s))\rangle$ for $s \in S$. The canonical morphism $p_X : X \to (X+Y)/\theta$ maps $x \in X$ to the θ-equivalence class of $\iota_1(x)$, and p_Y is defined analogously.

A weak pullback based on a relation $R \subseteq X \times Y$ is also an ordinary pullback, as one can derive from the joint injectivity of the two projections. Moreover, in Set pullbacks are jointly injective and pushouts are jointly surjective.

An object 0 of a category is called *initial* if for every other object X there exists precisely one arrow $! : 0 \to X$. Dually, an object 1 of a category is called *final* if for every object X there exists precisely one arrow $! : X \to 1$. Hence, the final object is the limit of the empty diagram, and the initial object is the colimit of the empty diagram. In Set the only initial object is the empty set and the final objects are the singleton sets. When we talk about an arbitrary final set, we denote its single element by a star, i.e. $1 = \{*\}$.

A functor $\mathcal{F}$ is said to *preserve* a (weak) (co)limit $\langle S, (s_i)_{i\in I}\rangle$ of a diagram D with objects $(D_i)_{i\in I}$ and arrows $(f_k)_{k\in K}$, if $\langle \mathcal{F}S, (\mathcal{F}s_i)_{i\in I}\rangle$ is again a (weak) (co)limit of the diagram with objects $(\mathcal{F}D_i)_{i\in I}$ and arrows $(\mathcal{F}f_k)_{k\in K}$, i.e. if it transforms a (weak) (co)limit of the diagram into a (weak) (co)limit of the transformed diagram. The functor $\mathcal{F}$ *weakly preserves* a (co)limit of a diagram, if it transforms it into a weak (co)limit of the transformed diagram.

For the sequel, the preservation of (weak) pullbacks will be of significant importance. We note the following two properties taken from [Gum99, Gum01].

Lemma 3.3.4. *In* Set*, a functor* $\mathcal{F}$ *preserves weak pullbacks if and only if it weakly preserves pullbacks.* □

Lemma 3.3.5. *A* Set *endofunctor* $\mathcal{F}$ *preserves weak pullbacks if and only if for any cospan* $\langle C, c_1 : X \to C, c_2 : Y \to C\rangle$ *we have: Given* u *and* v *with* $\mathcal{F}c_1(u) = \mathcal{F}c_2(v)$ *then there exists a* $w \in \mathcal{F}\{\langle x, y\rangle \mid c_1(x) = c_2(y)\}$ *with* $\mathcal{F}\pi_1(w) = u$ *and* $\mathcal{F}\pi_2(w) = v$*.* □

We end this section by mentioning a special type of pullback. A (weak) pullback is said to be *total* if its canonical morphisms are epi. In Set a pullback of a cospan $\langle C, c_1, c_2\rangle$ where $c_1 : X \to C$ and $c_2 : Y \to C$ are surjective, is a total pullback. Moreover, it is easy to see the following.

Lemma 3.3.6. *In* Set*, the pullback of a cospan* $\langle C, c_1 : X \to C, c_2 : Y \to C\rangle$ *is total if and only if the images of* X *and* Y *under* c_1 *and* c_2*, respectively, are equal, i.e.* $c_1(X) = c_2(Y)$. □

We say that a functor weakly preserves total pullbacks if it transforms any total pullback into a weak pullback. According to Lemma 3.3.6 weakly preserving total pullbacks is the same as weakly preserving pullbacks of cospans $\langle C, c_1, c_2\rangle$ with $c_1(X) = c_2(Y)$. Clearly, if a functor preserves weak pullbacks then it weakly preserves total pullbacks. We shall see in Chapter 5 that weak preservation of total pullbacks is a strictly weaker notion, i.e., an example of a functor will be given that weakly preserves total pullbacks but does not preserve weak pullbacks.

We note several facts of limits and colimits in $\mathsf{Coalg}_{\mathcal{F}}$ (cf. [Rut00]). All colimits exist in $\mathsf{Coalg}_{\mathcal{F}}$. They are constructed in the same way as in Set and then provided with a transition structure in a unique way. Limits in $\mathsf{Coalg}_{\mathcal{F}}$ exist provided the functor $\mathcal{F}$ preserves that type of limit. In that case, the limit is constructed the same way as in Set and provided with a transition structure in a unique way.

3.4 Coalgebraic bisimulations

One is often interested in the states of a coalgebra, i.e. the elements of its carrier set, only up to some sort of behavioral equivalence. The most common behavioral equivalence is *bisimilarity*. The definition is due to Aczel and Mendler [AM89].

Definition 3.4.1. *A* bisimulation *between two* $\mathcal{F}$*-coalgebras* $\langle S, \alpha\rangle$ *and* $\langle T, \beta\rangle$ *is a relation* $R \subseteq S \times T$ *such that there exists a coalgebra structure* $\gamma : R \to \mathcal{F}R$ *making the projections* $\pi_1 : R \to S$ *and* $\pi_2 : R \to T$ *coalgebra homomorphisms between the respective coalgebras, i.e. the two squares in the following diagram commute:*

$$\begin{array}{ccccc} S & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & T \\ {\scriptstyle\alpha}\downarrow & & {\scriptstyle\exists\gamma}\,\vdots & & \downarrow{\scriptstyle\beta} \\ \mathcal{F}S & \xleftarrow[\mathcal{F}\pi_1]{} & \mathcal{F}R & \xrightarrow[\mathcal{F}\pi_2]{} & \mathcal{F}T \end{array}$$

Occasionally we refer to γ *as the mediating coalgebra structure. We say that two states* $s \in S$ *and* $t \in T$ *are* bisimilar*, and write* $s \sim t$*, if they are related by some bisimulation between* $\langle S, \alpha\rangle$ *and* $\langle T, \beta\rangle$.

Example 3.4.2. Consider the transition systems, i.e. the $\mathcal{P}$-coalgebras. One can show, following the proof for LTSs ([RT93, Rut00]), that R is a bisimulation between two transition systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if

$$\langle s, t\rangle \in R \implies$$
$$1.\ s \to s' \implies (\exists t')\, t \to t' \text{ and } \langle s', t'\rangle \in R$$

$$2.\ t \to t' \implies (\exists s')s \to s' \text{ and } \langle s', t' \rangle \in R.$$

The conditions 1. and 2. from Example 3.4.2 are called *transfer conditions*. As we have seen, all the concrete definitions of bisimulation in Chapter 2 are expressed in terms of transfer conditions. In the case of transition systems (functor $\mathcal{P}$), it is not difficult to obtain transfer conditions for coalgebraic bisimulation. However, for more complex systems, it might be difficult to find, or just prove, the right transfer conditions for bisimulation. The next two sections of this chapter will be devoted to a method for characterizing bisimilarity in terms of transfer conditions.

A bisimulation relation on a coalgebra $\langle S, \alpha \rangle$ is any bisimulation between $\langle S, \alpha \rangle$ and itself. A bisimulation equivalence is a bisimulation on a coalgebra that is also an equivalence.

We next list some properties of bisimulation relations and bisimilarity. The proofs and more details can be found in [Rut00]. We group the properties in two lemmas, general properties and properties conditioned by weak pullback preservation.

Lemma 3.4.3. *The following properties hold:*

(i) *The diagonal $\Delta_S = \{\langle s, s \rangle \mid s \in S\}$ is a bisimulation equivalence on any coalgebra with state set S.*

(ii) *If R is a bisimulation between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$, then R^{-1} is a bisimulation between $\langle T, \beta \rangle$ and $\langle S, \alpha \rangle$.*

(iii) *Union of bisimulation relations is a bisimulation relation.*

(iv) *Bisimilarity is the largest bisimulation between two coalgebras.*

(v) *For any $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ and $\langle U, \gamma \rangle$, and any coalgebra homomorphisms $f : T \to S$, $g : T \to U$, the image $(f, g)T = \{\langle f(t), g(t) \rangle \mid t \in T\}$ is a bisimulation.* □

Lemma 3.4.4. *Assume that the functor $\mathcal{F}$ preserves weak pullbacks. Then:*

(i) *For any $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ and $\langle U, \gamma \rangle$, and any coalgebra homomorphisms $f : S \to T$, $g : U \to T$, the Set pullback $\langle P, \pi_1, \pi_2 \rangle$ of the cospan $\langle T, f, g \rangle$ is a bisimulation.*

(ii) *The relational composition of two bisimulations is again a bisimulation.*

(iii) *Bisimilarity on a coalgebra is an equivalence.* □

As a consequence of Lemma 3.4.3 and Lemma 3.4.4, we get that for weak pullback preserving functors, bisimilarity on a coalgebra is a union of all *equivalence* bisimulations on that coalgebra.

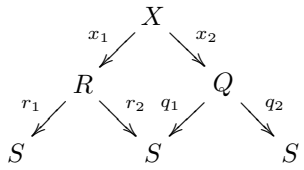
We will show that the statement of Lemma 3.4.4(iii) holds even if the assumption on the functor is relaxed to weakly preserving total pullbacks. For that purpose we formulate weaker properties corresponding to the properties of Lemma 3.4.4(i) and (ii), in the next two lemmas.

Lemma 3.4.5. *Let $\langle S, \alpha\rangle$, $\langle T, \beta\rangle$ and $\langle U, \gamma\rangle$ be $\mathcal{F}$-coalgebras, and $f : S \to T$, $g : U \to T$ coalgebra homomorphisms. If $\mathcal{F}$ weakly preserves total pullbacks and $f(S) = g(U)$, then the* Set *pullback $\langle P, \pi_1, \pi_2\rangle$ of the cospan $\langle T, f, g\rangle$ is a bisimulation.*

Proof In the proof of Lemma 3.4.4(i) (see [Rut00, Theorem 4.3]) in order to obtain that the pullback is a bisimulation, it is only used that $\mathcal{F}$ weakly preserves the pullback of the cospan $\langle T, f, g\rangle$. Now, by Lemma 3.3.6, if $\mathcal{F}$ weakly preserves total pullbacks, then it will weakly preserve this pullback, since additionally $f(S) = g(U)$. Hence the property holds. □

Lemma 3.4.6. *If the functor weakly preserves total pullbacks, then the relational composition of two reflexive bisimulations on one system is again a bisimulation.*

Proof The proof follows the same lines as the proof of Lemma 3.4.4(ii) ([Rut00, Theorem 5.4]). It is easy to see that given two relations R and Q on S, $R \circ Q = (r_1 \circ x_1, q_2 \circ x_2)(X)$ where X is the pullback of $\langle S, r_2, q_1\rangle$ as in the next diagram



and x_i, r_i, q_i denote the corresponding projections. The cospan $\langle S, r_2, q_1\rangle$ has the property $r_2(R) = S = q_1(Q)$ since R and Q are reflexive relations. Since $\mathcal{F}$ weakly preserves total pullbacks, by Lemma 3.4.5, we get that X is a bisimulation, i.e., the projections x_1 and x_2 are homomorphisms. Hence, $r_1 \circ x_1$ and $q_2 \circ x_2$ are also homomorphisms. Now, by Lemma 3.4.3(v), $R \circ Q$ is a bisimulation on $\langle S, \alpha\rangle$. □

We can now prove the analogue of Lemma 3.4.4(iii) in case of weak preservation of total pullbacks.

Lemma 3.4.7. *If the functor weakly preserves total pullbacks, then the bisimilarity relation on a coalgebra is an equivalence.*

Proof By Lemma 3.4.3(i-iv) the bisimilarity relation is reflexive and symmetric. Let $s_1 \sim s_2$ and $s_2 \sim s_3$ in an $\mathcal{F}$-coalgebra $\langle S, \alpha\rangle$. Let R and Q be two bisimulation relations such that $\langle s_1, s_2\rangle \in R$ and $\langle s_2, s_3\rangle \in Q$. By Lemma 3.4.3(i) and (iii) we can assume that R and Q are reflexive. Then, by Lemma 3.4.6, $R \circ Q$ is a bisimulation, and $\langle s_1, s_3\rangle \in R \circ Q$, from which we get $s_1 \sim s_3$. □

Hence, by Lemma 3.4.3, Lemma 3.4.6 and Lemma 3.4.7, also for functors that weakly preserve total pullbacks, the bisimilarity on a coalgebra is a union of all *equivalence* bisimulations on that coalgebra. A motivation for generalizing to functors that weakly preserve total pullbacks is that in Chapter 5 we will deal with a functor that weakly preserves total pullbacks but does not preserve weak pullbacks.

Recall that in Definition 3.3.3 by $\mathcal{T}_\tau$ we denoted the functor from $\mathcal{F}$-coalgebras to $\mathcal{G}$-coalgebras, induced by a natural transformation $\tau : \mathcal{F} \Longrightarrow \mathcal{G}$. Since $\mathcal{T}_\tau$ preserves homomorphisms, it also preserves bisimulations. This implies that if two states $s \in S$ and $t \in T$ are bisimilar in the $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ then they are also bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau \langle S, \alpha \rangle$ and $\mathcal{T}_\tau \langle T, \beta \rangle$.

3.5 Examples of coalgebras

In this section we present some more examples of coalgebras. In fact, we define almost all the functors whose coalgebras we will work with in later chapters in terms of basic and derived, or composed, functors. Furthermore, we show that both the basic and the composed functors preserve weak pullbacks. Hence, the bisimilarity for the coalgebras that we will consider is an equivalence.

Basic and composed functors

The basic functors in our consideration are the identity functor $\mathcal{I}d$, the constant functor $\underline{A}$ for a given set A, the powerset functor $\mathcal{P}$, the constant exponent functor $\mathcal{I}d^A$ (all of them defined in Example 3.1.2), as well as the probability distribution (cf. Definition 2.1.1) functor $\mathcal{D}$ defined below.

A coalgebra for the functor $\mathcal{I}d$ is a pair $\langle S, \alpha : S \to S \rangle$. Hence these are (unlabelled) fully deterministic systems, for which from any state there is a single next state. We will sometimes denote $s \to s'$ whenever $\alpha(s) = s'$.

An $\underline{A}$-coalgebra is a pair $\langle S, \alpha : S \to A \rangle$. It is not really a dynamic system, in fact it is a static labelling (or coloring) of a set of states.

As we already saw in Example 3.2.2, the $\mathcal{P}$ coalgebras $\langle S, \alpha : S \to \mathcal{P}S \rangle$ are the transition systems.

The $\mathcal{I}d^A$ - coalgebras are the deterministic labelled transition systems $\langle S, \alpha : S \to S^A \rangle$. For any state $s \in S$ in such a coalgebra, $\alpha(s)$ is a function from A to S. We can say that for any $a \in A$ there is a transition outgoing from s labelled by a to the unique next state $\alpha(s)(a)$. We use the transition notation $s \xrightarrow{a} s'$ to denote that $\alpha(s)(a) = s'$. These systems are deterministic since for any label a and any state s the next state is uniquely determined.

Definition 3.5.1. *The probability distribution functor*

$$\mathcal{D} : \mathsf{Set} \to \mathsf{Set}$$

maps a set S to

$$\mathcal{D}S = \{\mu : S \to \mathbb{R}_0^+ \mid \mu[S] = 1\}$$

and a function $f : S \to T$ to

$$(\mathcal{D}f)(\mu) = \ y \mapsto \mu[f^{-1}(y)]$$

where for a function $\mu : S \to \mathbb{R}_0^+$ and a subset $S' \subseteq S$ we write

$$\mu[S'] = \sum_{s \in S'} \mu(s).$$

The $\mathcal{D}$-coalgebras are exactly the Markov chains (see Definition 2.1.12). Let $\langle S, \alpha : S \to \mathcal{D}S\rangle$ be a $\mathcal{D}$-coalgebra. We introduce a transition notation as for Markov chains. We write $s \rightsquigarrow \mu$ whenever $\alpha(s) = \mu \in \mathcal{D}S$.

In the literature, the definition of the probability distribution functor is often restricted to probability distributions with finite support. These are functions $\mu : S \to \mathbb{R}_0^+$ such that $\mu[S] = 1$ and additionally, the support set

$$\mathrm{supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$$

is required to be finite. However, for our purposes, there is no need of restricting to finite support distributions.

Remark 3.5.2. Let us recall that the sum of an arbitrary family $\{x_i \mid i \in I\}$ of non-negative real numbers is defined as

$$\sum_{i \in I} x_i = \sup\{ \sum_{i \in J} x_i \mid J \subseteq I, J \text{ finite } \}.$$

Note that if $\sum_{i \in I} x_i$ is finite, then the support set $\{ x_i \mid x_i > 0 \}$ is at most countable (Proposition 2.1.2). Hence, the probability distributions have at most countable support set. Therefore they are sometimes also called *discrete* probability distributions.

From the basic functors we build composed functors, by the following three constructs: composition $\mathcal{F} \circ \mathcal{G}$, product $\mathcal{F} \times \mathcal{G}$ and coproduct $\mathcal{F} + \mathcal{G}$. Let $\mathcal{F}$ and $\mathcal{G}$ be Set endofunctors.

- The functor $\mathcal{F} \circ \mathcal{G}$ acts as applying first the functor $\mathcal{G}$ and then the functor $\mathcal{F}$. It maps a set S to the set $\mathcal{F}(\mathcal{G}S)$ and a function $f : S \to T$ to the function $\mathcal{F}(\mathcal{G}f)$.
- The functor $\mathcal{F} \times \mathcal{G}$ maps a set S to the product $\mathcal{F}S \times \mathcal{G}S$ and a function $f : S \to T$ to the function $\mathcal{F}f \times \mathcal{G}f$. Recall that the product of two functions $f' : S \to S'$ and $g' : T \to T'$ is the function $f' \times g' : S \times T \to S' \times T'$ given by $(f' \times g')\langle s, t\rangle = \langle f'(s), g'(t)\rangle$.

- The functor $\mathcal{F}+\mathcal{G}$ maps a set S to the coproduct $\mathcal{F}S+\mathcal{G}S$ and a function $f : S \to T$ to the function $\mathcal{F}f + \mathcal{G}f = [\iota_1 \circ \mathcal{F}f, \iota_2 \circ \mathcal{G}f]$. Recall that the case analysis of two functions $f : S \to U, g : T \to U$ is the function $[f, g] : S + T \to U$ defined by $[f, g](\iota_1(s)) = f(s)$ and $[f, g](\iota_2(t)) = g(t)$ for $s \in S$ and $t \in T$.

We will use the notation $\mathcal{F}^A$ for the composition of $\mathcal{I}d^A$ with a functor $\mathcal{F}$, i.e $\mathcal{F}^A = \mathcal{I}d^A \circ \mathcal{F}$.

Example 3.5.3. The fully deterministic systems from Example 3.2.5 were introduced as coalgebras of a functor $\times_A$ arising from the bifunctor $\times$. We can write $\times_A$ as a composed functor in the following way

$$\times_A = \underline{A} \times \mathcal{I}d.$$

We have seen in Example 3.2.3 that the labelled transition systems are coalgebras of the functor $\mathcal{B}$. Having defined the compositions of functors it is obvious that

$$\mathcal{B} = \mathcal{P} \circ (\underline{A} \times \mathcal{I}d) = \mathcal{P} \circ \times_A.$$

The LTSs can equivalently be considered as coalgebras of the functor $\mathcal{P}^A$.

Furthermore, almost all of the probabilistic systems from Chapter 2, are also coalgebras of composed functors. For example, the simple Segala systems are coalgebras of the functor $\mathcal{P} \circ (\underline{A} \times \mathcal{D}) = \mathcal{B} \circ \mathcal{D}$.

Weak pullback preservation

The following property is easy to show, for example by applying Lemma 3.3.5.

Lemma 3.5.4. *The functors $\mathcal{I}d$, $\underline{A}$, $\mathcal{P}$ and $\mathcal{I}d^A$ preserve weak pullbacks.* □

Next we establish the weak pullback preservation of $\mathcal{D}$. For the distribution functor with *finite support*, weak pullback preservation was proved by De Vink and Rutten [VR99], using the graph theoretic min cut - max flow theorem, and by Moss [Mos99], using an elementary matrix fill-in property. Following Moss [Mos99] we show that the needed matrix fill-in property can be used and holds for arbitrary, infinite, matrices as well. For the sake of completeness we give the proof in full detail.

Lemma 3.5.5. *Let A and B be sets and let $\phi : A \to \mathbb{R}_0^+$ and $\psi : B \to \mathbb{R}_0^+$ be such that*

$$\sum_{x \in A} \phi(x) = \sum_{y \in B} \psi(y) < \infty \tag{3.5}$$

Then there exists a function $\mu : A \times B \to \mathbb{R}_0^+$ such that for any $x \in A$ and any $y \in B$

$$\sum_{y \in B} \mu(x, y) = \phi(x), \qquad \sum_{x \in A} \mu(x, y) = \psi(y). \tag{3.6}$$

Before we present the rather technical proof, let us discuss the idea, also used in [Mos99], on a finite example. Let $A = \{1, 2, 3\}$, $B = \{1, 2, 3, 4\}$ and let ϕ and ψ be given by $\phi_1 = 2, \phi_2 = 1, \phi_3 = 3$ and $\psi_1 = 1, \psi_2 = 3, \psi_3 = 0, \psi_4 = 2$. Since

$$\phi_1 + \phi_2 + \phi_3 = \psi_1 + \psi_2 + \psi_3 + \psi_4$$

the statement claims that there exists a matrix M, in this case of order 3×4, such that ϕ_i is the sum of the i-th row and ψ_j the sum of the j-th column. The matrix

$$M = \begin{pmatrix} 1\,1\,0\,0 \\ 0\,1\,0\,0 \\ 0\,1\,0\,2 \end{pmatrix}$$

satisfies that property. We have constructed it in the following way. For $M(1,1)$ we take the minimum $\min\{\phi_1, \psi_1\}$, hence $M(1,1) = \psi_1 = 1$. Since the first column sum has already been achieved we fill-in $M(2,1) = M(3,1) = 0$ and the next element to be filled-in is $M(1,2)$. We fill it with the value $\min\{\phi_1 - M(1,1), \psi_2\} = \phi_1 - M(1,1) = 1$. Since the first row-sum has been achieved, we put $M(1,3) = M(1,4) = 0$, and continue with $M(2,2)$. It gets the value $\min\{\phi_2 - M(2,1), \psi_2 - M(1,2)\} = \phi_2 - M(2,1) = 1$. Hence, $M(2,3) = M(2,4) = 0$ and the next element to be filled-in is $M(3,2)$. Its value is then $\min\{\phi_3 - M(3,1), \psi_2 - M(1,2) - M(2,2)\} = \psi_2 - M(1,2) - M(2,2) = 1$, which completes the second column. Next is $M(3,3) = \min\{\phi_3 - M(3,1) - M(3,2), \psi_3 - M(1,3) - M(2,3)\} = \psi_3 - M(1,3) - M(2,3) = 0$. We fill-in the last element $M(3,4)$ with the remaining value $\phi_3 - M(3,1) - M(3,2) - M(3,3) = \psi_4 - M(1,4) - M(2,4) - M(3,4) = 2$.

Proof [of Lemma 3.5.5] We first consider the case when both A and B are countably infinite, i.e. we take $A = B = \mathbb{N}_0$. We recursively define a function

$$F : \mathbb{N} \to \mathbb{N}_0 \times \mathbb{N}_0 \times (\mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_0^+)$$

where $F(n) = \langle i(n), j(n), \mu_n \rangle$.

The idea behind the definition of F is that $F(n)$ represents the n-th iteration in the filling of the countable matrix. The pair $\langle i(n), j(n) \rangle$ represents the indices of the next entry that is to be filled in, and the function μ_n represents the n-th filling.

The function F is defined as follows. Put $F(1) = \langle 0, 0, \mu_1 \rangle$ for $\mu_1(i,j) = 0$ for all $i, j \in \mathbb{N}_0$. Assume $F(n)$ has already been defined. Put

$$\mu_{n+1}(i,j) = \begin{cases} \mu_n(i,j) & \text{if } \langle i,j \rangle \neq \langle i(n), j(n) \rangle \\ \min\{\phi(i(n)) - \sum_{k<j(n)} \mu_n(i(n),k), & \\ \quad \psi(j(n)) - \sum_{k<i(n)} \mu_n(k,j(n))\} & \text{if } \langle i,j \rangle = \langle i(n), j(n) \rangle \end{cases}$$

and

$$i(n+1) = \begin{cases} i(n)+1 & \mu_{n+1}(i(n),j(n)) = \phi(i(n)) - \sum_{j<j(n)} \mu_n(i(n),j) \\ i(n) & \text{otherwise} \end{cases}$$

$$j(n+1) = \begin{cases} j(n)+1 & \mu_{n+1}(i(n),j(n)) = \psi(j(n)) - \sum_{i<i(n)} \mu_n(i,j(n)) \\ j(n) & \text{otherwise} \end{cases}$$

Now, we define $\mu : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_0^+$ by

$$\mu(i,j) = \lim_{n\to\infty} \mu_n(i,j).$$

We will show that μ satisfies the requirements of the lemma. For this we first explore the properties of F.

It is obvious that F satisfies the following properties:

(a) $i(n+1) + j(n+1) > i(n) + j(n)$

(b) $i(n+1) \geq i(n)$

(c) $j(n+1) \geq j(n)$

(d) $\mu_{n+1}(i,j) = \mu_n(i,j),\ \langle i,j\rangle \neq \langle i(n), j(n)\rangle$

(e) $\mu_{n+1}(i,j) = 0,\ i > i(n)$ or $j > j(n)$

Hence, the indices of the next entry to be filled in, increase by one in at least one coordinate. Furthermore, μ_{n+1} differs from μ_n in at most one entry, the entry $\langle i(n), j(n)\rangle$ and $\mu_n(i,j)$ might be non-zero only in a finite initial rectangle.

We next show that F also satisfies the following properties:

$$\sum_{j\in\mathbb{N}_0} \mu_n(i,j) \begin{cases} = \phi(i) & i < i(n) \\ \leq \phi(i) & i \geq i(n) \end{cases} \tag{3.7}$$

$$\sum_{i\in\mathbb{N}_0} \mu_n(i,j) \begin{cases} = \psi(j) & j < j(n) \\ \leq \psi(j) & j \geq j(n) \end{cases} \tag{3.8}$$

The properties (3.7) and (3.8) indicate that μ_n is indeed an approximation of the required function. We prove (3.7), by induction on n. The proof of (3.8) is analogous. For $n = 1$ we have for all i, $i \geq i(1) = 0$ and

$$\sum_{j\in\mathbb{N}_0} \mu_1(i,j) = 0 \leq \phi(i).$$

Assume (3.7) holds for n. By (b), if $i < i(n)$ then $i < i(n+1)$, and by (d) we get

$$\sum_{j\in\mathbb{N}_0} \mu_{n+1}(i,j) = \sum_{j\in\mathbb{N}_0} \mu_n(i,j) \overset{(IH)}{=} \phi(i).$$

By the definition of F, if $i > i(n)$, then $i \geq i(n+1)$ and

$$\sum_{j\in\mathbb{N}_0} \mu_{n+1}(i,j) = \sum_{j\in\mathbb{N}_0} \mu_n(i,j) \overset{(IH)}{\leq} \phi(i).$$

The remaining case is $i = i(n)$. Then

$$\begin{aligned}\sum_{j\in\mathbb{N}_0}\mu_{n+1}(i,j) &= \left(\sum_{j<j(n)}\mu_{n+1}(i,j)\right) + \mu_{n+1}(i(n),j(n)) + \left(\sum_{j>j(n)}\mu_{n+1}(i,j)\right)\\ &\overset{(d)}{=} \left(\sum_{j<j(n)}\mu_{n}(i,j)\right) + \mu_{n+1}(i(n),j(n)) + \left(\sum_{j>j(n)}\mu_{n}(i,j)\right)\\ &\overset{(e)}{=} \sum_{j<j(n)}\mu_n(i,j) + \mu_{n+1}(i(n),j(n))\end{aligned}$$

By the definition of $\mu_{n+1}(i(n),j(n))$ we have

$$\mu_{n+1}(i(n),j(n)) \leq \phi(i(n)) - \sum_{j<j(n)}\mu_n(i(n),j) \tag{3.9}$$

Hence $\sum_{j\in\mathbb{N}_0}\mu_{n+1}(i,j) \leq \phi(i)$. Moreover, if $i < i(n+1)$ in (3.9) equality holds and thus also $\sum_{j\in\mathbb{N}_0}\mu_{n+1}(i,j) = \phi(i)$. This completes the proof of (3.7).

Now, we can show that F is well defined, i.e. that $\mu_n(i,j) \geq 0$ for all n,i,j, inductively on n. For $n = 1$ it is trivially satisfied. Assume that $\mu_n(i,j) \geq 0$, for all i,j. Then also $\mu_{n+1}(i,j) \geq 0$ for $\langle i,j\rangle \neq \langle i(n),j(n)\rangle$. By (3.7) and (3.8) we obtain

$$\phi(i(n)) \geq \sum_{j\in\mathbb{N}_0}\mu_n(i(n),j) \overset{(IH)}{\geq} \sum_{j<j(n)}\mu_n(i(n),j),$$

$$\psi(j(n)) \geq \sum_{i\in\mathbb{N}_0}\mu_n(i,j(n)) \overset{(IH)}{\geq} \sum_{i<i(n)}\mu_n(i,j(n))$$

and hence

$$\begin{aligned}&\mu_{n+1}(i(n),j(n)) =\\ &\qquad \min\{\phi(i(n)) - \sum_{j<j(n)}\mu_n(i(n),j), \psi(j(n)) - \sum_{i<i(n)}\mu_n(i,j(n))\} \geq 0.\end{aligned}$$

We next focus on the properties of the function μ. First of all note that the mapping $n \mapsto \langle i(n),j(n)\rangle$ is injective. Therefore, for any fixed index-pair $\langle i,j\rangle$, the sequence $(\mu_n(i,j))_{n\in\mathbb{N}}$ is either constantly 0, which happens if $\langle i,j\rangle \notin \{\langle i(n),j(n)\rangle \mid n \in \mathbb{N}\}$ or

$$\mu_n(i,j) = \begin{cases} 0 & n \leq n_0 \\ \mu_{n_0+1}(i,j) & n > n_0 \end{cases}$$

in case $\langle i,j\rangle = \langle i(n_0),j(n_0)\rangle$. In particular, we have established that the sequence $(\mu_n(i,j))_{n\in\mathbb{N}}$ is monotone, bounded, and it converges for any fixed (i,j), which makes $\mu : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_0^+$ well defined.

Finally, we show that μ satisfies the properties required in the assertion of the lemma. By (a) at least one of the sequences $(i(n))_{n\in\mathbb{N}}$, $(j(n))_{n\in\mathbb{N}}$ must tend to infinity, say $i(n)$ does. Let $i \in \mathbb{N}_0$ and let $n \in \mathbb{N}$ be such that $i < i(n)$. Then for all $m \geq n$ and all j,

$$\mu_m(i,j) = \mu_n(i,j) = \mu(i,j)$$

and thus, by (3.7)

$$\sum_{j\in\mathbb{N}_0} \mu(i,j) = \sum_{j\in\mathbb{N}_0} \mu_n(i,j) = \phi(i),$$

i.e. the first part of (3.6) holds true. It follows that

$$\sum_{j\in\mathbb{N}_0} \psi(j) = \sum_{i\in\mathbb{N}_0} \phi(i) = \sum_{i\in\mathbb{N}_0}\sum_{j\in\mathbb{N}_0} \mu(i,j) = \sum_{j\in\mathbb{N}_0}\sum_{i\in\mathbb{N}_0} \mu(i,j), \tag{3.10}$$

where the change in the order of summation is justified by the fact that $\mu_n(i,j) \geq 0$. Since, by (3.8), $\psi(j) \geq \sum_{i\in\mathbb{N}_0} \mu_n(i,j)$ for all n we obtain that

$$\sum_{i\in\mathbb{N}_0} \mu(i,j) = \lim_{n\to\infty} \sum_{i\in\mathbb{N}_0} \mu_n(i,j) \leq \psi(j). \tag{3.11}$$

Here, the change of the limit and the sum is allowed since $\mu_n(i,j)$ is a non-negative, monotone sequence. Now (3.10), (3.11), and the assumption $\sum_{j\in\mathbb{N}_0} \psi(j) < \infty$, imply that

$$\sum_{i\in\mathbb{N}_0} \mu(i,j) = \psi(j),\ j \in \mathbb{N}_0.$$

This completes the proof in the case $A = B = \mathbb{N}_0$.

Assume now that A, B, ϕ, ψ are as in the formulation of the lemma. Consider $A' = \{x \in A \mid \phi(x) \neq 0\}$, $B' = \{x \in B \mid \psi(x) \neq 0\}$, $\phi' = \phi|_{A'}$, $\psi' = \psi|_{B'}$. Then A' and B' are at most countable, for otherwise condition (3.5) would be violated. If $\mu' : A' \times B' \to \mathbb{R}_0^+$ is such that for any $x \in A', y \in B'$

$$\sum_{y\in B'} \mu'(x,y) = \phi(x), \quad \sum_{x\in A'} \mu'(x,y) = \psi(y)$$

then the function $\mu : A \times B \to \mathbb{R}_0^+$ defined by

$$\mu(x,y) = \begin{cases} \mu'(x,y) & \text{if } \langle x,y\rangle \in A' \times B' \\ 0 & \text{otherwise} \end{cases}$$

fulfills the requirements of the lemma. Hence, it is enough to consider the case when A and B are at most countable. Write $A = \{a_k \mid k \in \mathbb{N}_0, k < |A|\}$ and $B = \{b_l \mid l \in \mathbb{N}_0, l < |B|\}$ and define $\phi', \psi' : \mathbb{N}_0 \to \mathbb{R}_0^+$ by

$$\phi'(k) = \begin{cases} \phi(a_k) & \text{if } k < |A| \\ 0 & \text{otherwise} \end{cases} \qquad \psi'(l) = \begin{cases} \psi(b_l) & \text{if } l < |B| \\ 0 & \text{otherwise} \end{cases}$$

We obtain $\mu' : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_0^+$ with $\sum_{l\in\mathbb{N}_0} \mu'(k,l) = \phi'(k)$ and $\sum_{k\in\mathbb{N}_0} \mu'(k,l) = \psi'(l)$ for all $k, l \in \mathbb{N}_0$. If $k \geq |A|$ then $\phi'(k) = 0$ and hence $\mu'(k,l) = 0$ for $l \in \mathbb{N}_0$. Similarly, for $l \geq |B|$, $\mu'(k,l) = 0$ for $k \in \mathbb{N}_0$. Thus

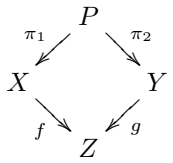
$$\mu(a_k, b_l) = \mu'(k,l),\ k < |A|,\ l < |B|$$

satisfies the requirements of the lemma. □

We are now ready to show the weak pullback preservation of the functor $\mathcal{D}$.

Lemma 3.5.6. *The functor $\mathcal{D}$ preserves weak pullbacks.*

Proof It suffices to show that a pullback diagram



will be transformed to a weak pullback diagram.

Let P' be the pullback of the cospan $\mathcal{D}X \xrightarrow{\mathcal{D}f} \mathcal{D}Z \xleftarrow{\mathcal{D}g} \mathcal{D}Y$. Let $\langle u, v\rangle \in P'$. Then $(\mathcal{D}f)(u) = (\mathcal{D}g)(v)$. By Lemma 3.3.5, we need to find $\mu \in \mathcal{D}P$ such that

$$(\mathcal{D}\pi_1)(\mu) = u, \qquad (\mathcal{D}\pi_2)(\mu) = v. \tag{3.12}$$

Since for $x \in X$ and $y \in Y$,

$$(\mathcal{D}\pi_1)(\mu) = x \mapsto \mu[\pi_1^{-1}(x)], \qquad (\mathcal{D}\pi_2)(\mu) = y \mapsto \mu[\pi_2^{-1}(y)]$$

condition (3.12) is equivalent to

$$\mu[\pi_1^{-1}(x)] = u(x), \qquad \mu[\pi_2^{-1}(y)] = v(y),$$

for $x \in X, y \in Y$, i.e.,

$$\sum_{y\in Y:\langle x,y\rangle\in P} \mu(x,y) = u(x), \qquad \sum_{x\in X:\langle x,y\rangle\in P} \mu(x,y) = v(y). \tag{3.13}$$

The set P can be written as the union

$$P = \bigcup_{z\in Z} f^{-1}(z) \times g^{-1}(z)$$

of disjoint rectangles. In fact these rectangles have non-overlapping edges in the following sense: Let $Z_1 = f^{-1}(z_1) \times g^{-1}(z_1)$, and $Z_2 = f^{-1}(z_2) \times g^{-1}(z_2)$, for $z_1, z_2 \in Z$. Then, if $\langle x, y\rangle \in Z_1$, it follows that $\langle x, y'\rangle, \langle x', y\rangle \notin Z_2$ for all $x' \in X, y' \in Y$, and vice-versa. Therefore, the existence of a map μ which

satisfies condition (3.13) is equivalent to the condition that for all $z \in Z$ there exists a function $\mu_z : f^{-1}(z) \times g^{-1}(z) \to \mathbb{R}_0^+$ such that for all $x \in f^{-1}(z)$, and all $y \in g^{-1}(z)$,

$$\sum_{y \in g^{-1}(z)} \mu_z(x,y) = u(x), \qquad \sum_{x \in f^{-1}(z)} \mu_z(x,y) = v(y).$$

Since $\langle u, v \rangle \in P'$, we have

$$\sum_{x \in f^{-1}(z)} u(x) = u[f^{-1}(z)] = (\mathcal{D}f)(u)(z) = (\mathcal{D}g)(v)(z) = v[g^{-1}(z)] = \sum_{y \in g^{-1}(z)} v(y).$$

Hence, we may apply the matrix-fill-in property, Lemma 3.5.5. □

We finish this section by stating the weak pullback preservation for the composed functors.

Lemma 3.5.7. *If $\mathcal{F}$ and $\mathcal{G}$ preserve weak pullbacks, then so do $\mathcal{F} \circ \mathcal{G}$, $\mathcal{F} \times \mathcal{G}$ and $\mathcal{F} + \mathcal{G}$.* □

The proof of Lemma 3.5.7 can be derived directly from the definitions, or from Lemma 3.3.5. Hence, any functor that can be built from our basic functors by applying composition, product and coproduct preserves weak pullbacks.

3.6 From coalgebraic bisimulation to transfer conditions

In this section we present a way of characterizing coalgebraic bisimulation and hence bisimilarity in terms of transfer conditions.

The procedure for characterizing bisimilarity is strongly connected to a notion of relation liftings (cf. [Jac02, JH03]). We will focus on relation liftings in more detail in the next subsection. For the moment, the following definition suffices.

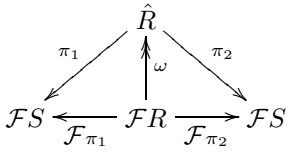
Definition 3.6.1. *Let $R \subseteq S \times T$ be a relation, and $\mathcal{F}$ a* Set *functor. The relation R can be lifted to a relation* $\mathrm{Rel}(\mathcal{F})(R) \subseteq \mathcal{F}S \times \mathcal{F}T$ *defined by*

$$\langle x, y \rangle \in \mathrm{Rel}(\mathcal{F})(R) \iff \exists z \in \mathcal{F}R\colon \mathcal{F}\pi_1(z) = x,\ \mathcal{F}\pi_2(z) = y.$$

It is easy to see that the following lemma holds.

Lemma 3.6.2. *Let $\mathcal{F}$ be a* Set *endofunctor. A relation $\hat{R} \subseteq \mathcal{F}S \times \mathcal{F}T$ is the lifting of a relation $R \subseteq S \times T$ if and only if there exists a surjective map $\omega : \mathcal{F}R \to \hat{R}$ such that the following diagram commutes.*

(3.14)



□

The surjective map $\omega : \mathcal{F}R \to \mathrm{Rel}(\mathcal{F})(R)$ is given by $\omega(z) = \langle \mathcal{F}\pi_1(z), \mathcal{F}\pi_2(z)\rangle$.

Remark 3.6.3. By comparing Definition 2.1.4 and Definition 3.6.1, we see that the distribution lifting $\equiv_R$ of a relation $R \subseteq S \times T$ is exactly the lifting $\mathrm{Rel}(\mathcal{D})(R)$.

The following lemma provides transfer conditions for bisimulation in terms of relation lifting.

Lemma 3.6.4. *A relation $R \subseteq S \times T$ is a bisimulation between the $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if*

$$\langle s, t\rangle \in R \implies \langle \alpha(s), \beta(t)\rangle \in \mathrm{Rel}(\mathcal{F})(R). \tag{3.15}$$

Proof Let R be a bisimulation between the $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$, and let $\langle s, t\rangle \in R$. Let γ be the mediating coalgebra structure for R. Then $\gamma(\langle s, t\rangle)$ satisfies $\mathcal{F}\pi_1(\gamma(\langle s, t\rangle)) = \alpha(s)$ and $\mathcal{F}\pi_2(\gamma(\langle s, t\rangle)) = \beta(t)$. Hence, by Definition 3.6.1, $\langle \alpha(s), \beta(t)\rangle \in \mathrm{Rel}(\mathcal{F})(R)$.

For the opposite, assume R satisfies condition (3.15). For $\langle s, t\rangle \in R$, put $\gamma(\langle s, t\rangle) = z$ for some z such that $\omega(z) = \langle \alpha(s), \beta(t)\rangle$. Such an element $z \in \mathcal{F}R$ exists since $\omega : \mathcal{F}R \to \mathrm{Rel}(\mathcal{F})(R)$ is surjective and $\langle \alpha(s), \beta(t)\rangle \in \mathrm{Rel}(\mathcal{F})(R)$. Then R is a bisimulation with mediating coalgebra structure γ. □

The right-hand side of condition (3.15) is already a transfer condition. We shall see in the later subsections how concrete transfer conditions can be instantiated out of it for our basic and composed functors. However, for some functors also the next observation will be helpful.

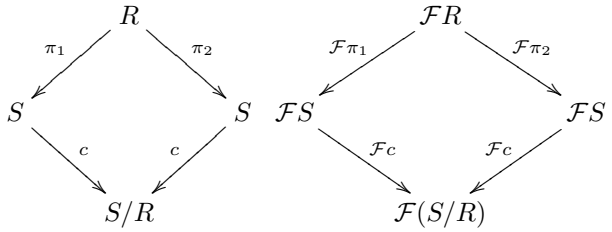
Recall that a functor weakly preserves total pullbacks if it transforms any pullback diagram with epi morphisms into a weak pullback diagram.

Lemma 3.6.5. *If the functor $\mathcal{F}$ weakly preserves total pullbacks and R is an equivalence on S, then* $\mathrm{Rel}(\mathcal{F})(R)$ *is the pullback in* Set *of the cospan*

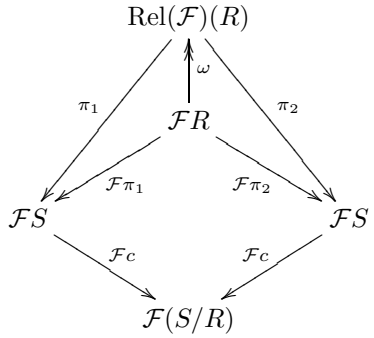
$$\mathcal{F}S \xrightarrow{\mathcal{F}c} \mathcal{F}(S/R) \xleftarrow{\mathcal{F}c} \mathcal{F}S \tag{3.16}$$

where $c\colon S \to S/R$ is the canonical morphism mapping each element to its equivalence class.

Proof Since R is an equivalence relation and therefore reflexive, the left diagram below is a pullback diagram with epi legs.



By the assumption, the right diagram is a weak pullback diagram. By (3.14) the surjective map $\omega\colon \mathcal{F}R \to \mathrm{Rel}(\mathcal{F})(R)$ makes the two upper triangles of the next diagram commutative:



Since ω is surjective the outer square of the above diagram also commutes, and by the existence of ω from the weak pullback $\mathcal{F}R$ to $\mathrm{Rel}(\mathcal{F})(R)$, $\mathrm{Rel}(\mathcal{F})(R)$ is a weak pullback as well. However, since it is based on a relation, it is a pullback (see page 72). □

By the construction of a pullback in Set, we can formulate the following corollary.

Corollary 3.6.6. *A relation $R \subseteq S \times S$ is an equivalence bisimulation on the $\mathcal{F}$-coalgebra $\langle S, \alpha\rangle$, where $\mathcal{F}$ weakly preserves total pullbacks, if and only if*

$$\langle s,t\rangle \in R \implies (\mathcal{F}c \circ \alpha)(s) = (\mathcal{F}c \circ \alpha)(t). \tag{3.17}$$

where $c\colon S \to S/R$ is the canonical morphism mapping each element to its R-equivalence class. □

Note that Corollary 3.6.6 provides transfer conditions for equivalence bisimulations only. This is still of use since often in the literature (cf. Chapter 2) for concrete systems, bisimulations are defined as equivalence relations. We have seen that the bisimilarity is a union of all equivalence bisimulations, for functors that weakly preserve total pullbacks. Therefore, also Corollary 3.6.6 provides a way to characterize bisimilarity in terms of transfer conditions, for functors that weakly preserve total pullbacks. It is handy in cases in which we can not immediately apply Lemma 3.6.4, as it will be the case in Chapter 5.

As an easy consequence of Lemma 3.6.5 for the distribution functor we obtain Proposition 2.1.5. Namely, let R be an equivalence relation on a set S. By Remark 3.6.3, $\equiv_R = \mathrm{Rel}(\mathcal{D})(R)$ and by Lemma 3.6.5 we get that $\equiv_R$ is the pullback of the cospan

$$\mathcal{D}S \xrightarrow{\mathcal{D}c} \mathcal{D}(S/R) \xleftarrow{\mathcal{D}c} \mathcal{D}S\ .$$

Hence,

$$\begin{aligned} \mu \equiv_R \nu \quad &\iff \quad \mathcal{D}c(\mu) = \mathcal{D}c(\nu) \\ &\iff \quad \text{for any } C \in S/R : \mu[c^{-1}(C)] = \nu[c^{-1}(C)] \\ &\iff \quad \text{for any } C \in S/R : \mu[C] = \nu[C]. \end{aligned}$$

3.6.1 Properties of relation liftings

In this section we focus in more detail on relation liftings. Most of the presented results are known and they can be found in the papers by Jacobs et al. [HJ98, Jac02, JH03, Jac04].

Recall the definition of the category Rel from Example 3.1.1. For an endofunctor $\mathcal{F}$ on Set, the relation lifting $\mathrm{Rel}(\mathcal{F})$ is an endofunctor $\mathrm{Rel}(\mathcal{F}) : \mathsf{Rel} \to \mathsf{Rel}$ that makes the following diagram commute

$$\begin{array}{ccc} \mathsf{Rel} & \xrightarrow{\mathrm{Rel}(\mathcal{F})} & \mathsf{Rel} \\ {\scriptstyle \mathcal{U}}\downarrow & & \downarrow{\scriptstyle \mathcal{U}} \\ \mathsf{Set} \times \mathsf{Set} & \xrightarrow{\mathcal{F} \times \mathcal{F}} & \mathsf{Set} \times \mathsf{Set} \end{array}$$

where $\mathcal{U} : \mathsf{Rel} \to \mathsf{Set} \times \mathsf{Set}$ is the forgetful functor that maps a relation $R \subseteq S \times T$ to its underlying sets $S \times T$, and an arrow in Rel is mapped to itself.

For functors that preserve weak pullbacks, the relation lifting preserves the following [Jac02, JH03, Jac04]

- equality: $\mathrm{Rel}(\mathcal{F})(\Delta_S) = \Delta_{\mathcal{F}S}$,
- relational composition: $\mathrm{Rel}(\mathcal{F})(Q \circ R) = \mathrm{Rel}(\mathcal{F})(Q) \circ \mathrm{Rel}(\mathcal{F})(R)$,
- inclusions: $R \subseteq Q \implies \mathrm{Rel}(\mathcal{F})(R) \subseteq \mathrm{Rel}(\mathcal{F})(Q)$,
- inverse relations: $\mathrm{Rel}(\mathcal{F})(R^{-1}) = \mathrm{Rel}(\mathcal{F})(R)^{-1}$,
- inverse images: for $R \subseteq S \times T$, $f : S' \to S$ and $g : T' \to T$ we have $\mathrm{Rel}(\mathcal{F})((f \times g)^{-1}(R)) = (\mathcal{F}f \times \mathcal{F}g)^{-1}(\mathrm{Rel}(\mathcal{F})(R))$.

Hence, the characteristic properties of equivalences and preorders are also preserved by relation liftings.

It has already been reported by Jacobs (cf. [Jac02]) that for the special case of polynomial functors $\mathrm{Rel}(\mathcal{F})$ may equivalently be defined inductively on the structure of $\mathcal{F}$. The polynomial functors form a subclass of our class of basic and derived functors. The next lemma shows that also for our class of inductively defined functors, $\mathrm{Rel}(\mathcal{F})$ can be defined by structural induction.

Lemma 3.6.7. *Let $R \subseteq S \times T$. Then:*

(*i*) $\mathrm{Rel}(\mathcal{I}d_{\mathsf{Set}}) = \mathcal{I}d_{\mathsf{Rel}}$.

(*ii*) $\mathrm{Rel}(\underline{A}) = \underline{\Delta_A}$ *i.e. the constant functor that maps any relation* R *to the diagonal relation* Δ_A.

(*iii*) $$\mathrm{Rel}(\mathcal{P})(R) = \left\{ \langle X, Y\rangle \mid (\forall x \in X)(\exists y \in Y)\langle x, y\rangle \in R \wedge (\forall y \in Y)(\exists x \in X)\langle x, y\rangle \in R \right\}.$$

(*iv*) $\mathrm{Rel}(\mathcal{I}d^A)(R) = \{\langle f, g\rangle \mid (\forall a \in A)\langle f(a), g(a)\rangle \in R\}$.

(*v*) $\mathrm{Rel}(\mathcal{D})(R) = \equiv_R$.

(*vi*) $\mathrm{Rel}(\mathcal{F} \circ \mathcal{G})(R) = \mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R))$.

(*vii*) $$\mathrm{Rel}(\mathcal{F} \times \mathcal{G})(R) = \left\{ \left\langle \langle x_1, x_2\rangle, \langle y_1, y_2\rangle \right\rangle \mid \langle x_1, y_1\rangle \in \mathrm{Rel}(\mathcal{F})(R) \wedge \langle x_2, y_2\rangle \in \mathrm{Rel}(\mathcal{G})(R) \right\}.$$

(*viii*) $$\begin{aligned}\mathrm{Rel}(\mathcal{F} + \mathcal{G})(R) = &\left\{ \langle \iota_1(x_1), \iota_1(y_1)\rangle \mid \langle x_1, y_1\rangle \in \mathrm{Rel}(\mathcal{F})(R) \right\} \cup \\ &\left\{ \langle \iota_2(x_2), \iota_2(y_2)\rangle \mid \langle x_2, y_2\rangle \in \mathrm{Rel}(\mathcal{G})(R) \right\}.\end{aligned}$$

Proof All of the listed properties can be proved from Definition 3.6.1 and using Lemma 3.6.2. We shall only present the proof of (*vi*). Let $R \subseteq S \times T$. By Lemma 3.6.2 we get that $\mathrm{Rel}(\mathcal{G})(R)$ is the unique relation such that there exists a surjective function $\omega_{\mathcal{G}}$ making the following diagram commute

$$\begin{array}{ccccc} & & \mathrm{Rel}(\mathcal{G})(R) & & \\ & \swarrow^{g_1} & \uparrow{\scriptstyle \omega_{\mathcal{G}}} & \searrow^{g_2} & \\ \mathcal{G}S & \xleftarrow[\mathcal{G}\pi_1]{} & \mathcal{G}R & \xrightarrow[\mathcal{G}\pi_2]{} & \mathcal{G}T \end{array} \tag{3.18}$$

where g_1 and g_2 denote the projections from $\mathrm{Rel}(\mathcal{G})(R)$ and π_1, π_2 the projections from R. By applying $\mathcal{F}$ to the diagram (3.18) we get the following commuting diagram.

$$\begin{array}{ccccc} & & \mathcal{F}(\mathrm{Rel}(\mathcal{G})(R)) & & \\ & \swarrow^{\mathcal{F}g_1} & \uparrow{\scriptstyle \mathcal{F}\omega_{\mathcal{G}}} & \searrow^{\mathcal{F}g_2} & \\ \mathcal{F}\mathcal{G}S & \xleftarrow[\mathcal{F}\mathcal{G}\pi_1]{} & \mathcal{F}\mathcal{G}R & \xrightarrow[\mathcal{F}\mathcal{G}\pi_2]{} & \mathcal{F}\mathcal{G}T \end{array} \tag{3.19}$$

On the other hand, by Lemma 3.6.2, $\mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R))$ is the unique relation

for which there exists a surjective mapping ω making the next diagram commute

$$\begin{array}{ccccc} & & \mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R)) & & \\ & \overset{f_1}{\swarrow} & \uparrow{\scriptstyle \omega} & \overset{f_2}{\searrow} & \\ \mathcal{FG}S & \xleftarrow[\mathcal{F}g_1]{} & \mathcal{F}(\mathrm{Rel}(\mathcal{G})(R)) & \xrightarrow[\mathcal{F}g_2]{} & \mathcal{FG}T \end{array} \tag{3.20}$$

where f_1 and f_2 are the projections from $\mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R))$. By plugging (3.19) and (3.20) together we get that the next diagram commutes

$$\begin{array}{ccccc} & & \mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R)) & & \\ & \overset{f_1}{\swarrow} & \uparrow{\scriptstyle \omega\circ\mathcal{F}\omega_{\mathcal{G}}} & \overset{f_2}{\searrow} & \\ \mathcal{FG}S & \xleftarrow[\mathcal{FG}\pi_1]{} & \mathcal{FG}R & \xrightarrow[\mathcal{FG}\pi_2]{} & \mathcal{FG}T \end{array}$$

and $\omega \circ \mathcal{F}\omega_{\mathcal{G}}$ is a surjective map. Hence, by Lemma 3.6.2, $\mathrm{Rel}(\mathcal{F})(\mathrm{Rel}(\mathcal{G})(R)) = \mathrm{Rel}(\mathcal{F} \circ \mathcal{G})(R)$. □

3.6.2 Transfer conditions for the basic and the composed functors

Having the inductive definition of $\mathrm{Rel}(\mathcal{F})(R)$ from Lemma 3.6.7 and the characterization of Lemma 3.6.4 we now spell out the transfer conditions for bisimulation relations for our set of basic and composed functors.

Let $R \subseteq S \times T$. Let $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ be $\mathcal{F}$-coalgebras. The relation R is a bisimulation between $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if whenever $\langle s, t\rangle \in R$ then the transfer condition $\langle \alpha(s), \beta(t)\rangle \in \mathrm{Rel}(\mathcal{F})(R)$ holds. When $\mathcal{F}$ is instantiated to one of our basic functors, by Lemma 3.6.7 and Lemma 3.6.4 we get the following concrete transfer conditions.

$\mathcal{I}d$ **coalgebras**:

By Lemma 3.6.7(i) the transfer condition is $\langle \alpha(s), \beta(t)\rangle \in R$, i.e.

$$\text{if } s \to s' \text{ and } t \to t', \quad \text{then } \langle s', t'\rangle \in R. \tag{3.21}$$

Note that the largest equivalence on S, $\nabla_S = S \times S$ is a bisimulation on an $\mathcal{I}d$-coalgebra $\langle S, \alpha\rangle$. Therefore for the bisimilarity we have $\sim \; = \; \nabla_S$, i.e., any two states in an $\mathcal{I}d$-coalgebra are bisimilar.

$\underline{A}$ **coalgebras**:

The transfer condition for bisimulation of $\underline{A}$-coalgebras is

$$\alpha(s) =_A \alpha(t). \tag{3.22}$$

In other words, bisimilar states have the same label or color.

$\mathcal{P}$ coalgebras:

For transition systems we get the following transfer condition:

$$(\forall s' \in \alpha(s), \exists t' \in \beta(t) : \ \langle s', t' \rangle \in R) \ \wedge \ (\forall t' \in \beta(t), \exists s' \in \alpha(s) : \ \langle s', t' \rangle \in R) \, .$$

In terms of transitions, the condition reads

$$\begin{array}{l} \text{if } s \to s', \text{ then there exists } t' \text{ with } t \to t' \text{ and } \langle s', t' \rangle \in R, \text{ and} \\ \text{if } t \to t', \text{ then there exists } s' \text{ with } s \to s' \text{ and } \langle s', t' \rangle \in R. \end{array} \tag{3.23}$$

$\mathcal{I}d^A$ coalgebras:

The transfer condition is $\forall a \in A : \langle \alpha(s)(a), \beta(t)(a) \rangle \in R$, i.e.

$$\text{for all } \ a \in A : \ \text{if } \ s \xrightarrow{a} s' \text{ and } \ t \xrightarrow{a} t', \ \text{then } \ \langle s', t' \rangle \in R. \tag{3.24}$$

$\mathcal{D}$ coalgebras:

For Markov chains the transfer condition is $\langle \alpha(s), \beta(t) \rangle \ \in \ \equiv_R$, i.e.

$$\text{if } \ s \rightsquigarrow \mu \text{ and } \ t \rightsquigarrow \nu, \ \text{then } \ \mu \equiv_R \nu. \tag{3.25}$$

Note that the condition (3.25) is the same as the one given by Definition 2.1.19. Hence, we have shown that a relation is a bisimulation (equivalence) according to Definition 3.4.1 if and only if it is a bisimulation on Markov chains, according to Definition 2.1.19. As a consequence, coalgebraic bisimilarity coincides with the concrete bisimilarity in the case of Markov chains. This is an alternative proof to the result of De Vink and Rutten [VR99] for Markov chains. Also for $\mathcal{D}$ systems, any two states in a $\mathcal{D}$-coalgebra are bisimilar.

Derived functors

Next we write down the transfer conditions for the derived functors.

$\mathcal{F} \times \mathcal{G}$ In this case we have that the transition structures α and β are pairings $\alpha = (\alpha_{\mathcal{F}}, \alpha_{\mathcal{G}})$ and $\beta = (\beta_{\mathcal{F}}, \beta_{\mathcal{G}})$ where $\alpha_{\mathcal{F}}, \beta_{\mathcal{F}}$ are transition structures of an $\mathcal{F}$-coalgebra, and $\alpha_{\mathcal{G}}, \beta_{\mathcal{G}}$ are transition structures of an $\mathcal{G}$-coalgebra. By Lemma 3.6.7(vii), the transfer condition is:

$$\langle \alpha_{\mathcal{F}}(s), \beta_{\mathcal{F}}(t) \rangle \in \mathrm{Rel}(\mathcal{F})(R) \ \wedge \ \langle \alpha_{\mathcal{G}}(s), \beta_{\mathcal{G}}(t) \rangle \in \mathrm{Rel}(\mathcal{G})(R). \tag{3.26}$$

$\mathcal{F} + \mathcal{G}$ Now, we get a transfer condition

$$\begin{array}{l} \alpha(s) = \iota_1(x), \beta(t) = \iota_1(y), \ \langle x, y \rangle \in \mathrm{Rel}(\mathcal{F})(R) \ \vee \\ \alpha(s) = \iota_2(x), \beta(t) = \iota_2(y), \ \langle x, y \rangle \in \mathrm{Rel}(\mathcal{G})(R). \end{array}$$

or, if we omit the injections and consider the coproduct as a disjoint union:

$$\begin{array}{ll} \text{either} & \alpha(s) \in \mathcal{F}S \text{ and } \ \beta(t) \in \mathcal{F}T \text{ and } \ \langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{F})(R) \\ \text{or} & \alpha(s) \in \mathcal{G}S \text{ and } \beta(t) \in \mathcal{G}T \text{ and } \ \langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{G})(R). \end{array} \tag{3.27}$$

We can not directly write down a transfer condition for $\mathcal{F} \circ \mathcal{G}$ in general, but having the transfer conditions for $\mathcal{F}$ and for $\mathcal{G}$, by Lemma 3.6.7(vi) we can derive transfer conditions in particular cases.

The inductive construction of transfer conditions for labelled transition systems is illustrated in the next example.

Example 3.6.8. The functor defining LTSs is $\mathcal{B} = \mathcal{P} \circ (\underline{A} \times \mathcal{I}d)$. A relation $R \subseteq S \times T$ is a bisimulation between the two LTSs $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ if and only if

$$\langle s, t \rangle \in R \implies \langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{B})(R).$$

Using the inductive definition we get, by Lemma 3.6.7(vi),

$$\langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{B})(R)$$

if and only if

$$\begin{array}{l} (\forall s' \in \alpha(s), \exists t' \in \beta(t) : \ \langle s', t' \rangle \in \mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)) \ \wedge \\ (\forall t' \in \alpha(t), \exists s' \in \beta(s) : \ \langle s', t' \rangle \in \mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)) \end{array}$$

which by Lemma 3.6.7(vii) and (3.26) is equivalent to

$$\begin{array}{l} \big(\forall \langle a, s' \rangle \in \alpha(s), \exists \langle b, t' \rangle \in \beta(t) : \ \langle a, b \rangle \in \mathrm{Rel}(\underline{A})(R) \ \wedge \ \langle s', t' \rangle \in \mathrm{Rel}(\mathcal{I}d)(R)\big) \ \wedge \\ \big(\forall \langle a, t' \rangle \in \beta(t), \exists \langle b, s' \rangle \in \alpha(s) : \ \langle a, b \rangle \in \mathrm{Rel}(\underline{A})(R) \ \wedge \ \langle s', t' \rangle \in \mathrm{Rel}(\mathcal{I}d)(R)\big). \end{array}$$

Applying Lemma 3.6.7(i),(ii) we get the following equivalent condition

$$\begin{array}{l} \big(\forall \langle a, s' \rangle \in \alpha(s), \exists \langle b, t' \rangle \in \beta(t) : \ a = b \ \wedge \ \langle s', t' \rangle \in R\big) \ \wedge \\ \big(\forall \langle a, t' \rangle \in \beta(t), \exists \langle b, s' \rangle \in \alpha(s) : \ a = b \ \wedge \ \langle s', t' \rangle \in R\big). \end{array}$$

Finally, we rewrite the last condition in terms of transition notation and obtain the transfer condition for bisimulation between labelled transition systems:

$$\begin{array}{l} \text{if } s \xrightarrow{a} s', \text{ then there exists } t' \text{ with } t \xrightarrow{a} t' \text{ and } \langle s', t' \rangle \in R, \text{ and} \\ \text{if } t \xrightarrow{a} t', \text{ then there exists } s' \text{ with } s \xrightarrow{a} s' \text{ and } \langle s', t' \rangle \in R. \end{array} \tag{3.28}$$

Hence, as it is well-known [RT93, Rut00], coalgebraic bisimilarity coincides with concrete bisimilarity (Definition 2.1.14) for the case of labelled transition systems.

4

A hierarchy of probabilistic system types

In this chapter we arrange the various classes of probabilistic systems in an expressiveness hierarchy. We model the different system types as coalgebras of suitable behavior functors and argue that the corresponding coalgebraic bisimilarity coincides with concrete probabilistic bisimilarity. The theory of coalgebras provides a unifying framework for the presentation of the various classes and the system translations we needed to establish the hierarchy. All these translations arise in a standard way from natural transformations between the two behavior functors involved.

Probabilistic systems of different kinds have been studied as semantic objects since the early nineties. Some of them arise from nondeterministic systems by adding probabilistic information to all choices; sometimes both types of uncertainty are mixed. The main motivation for considering probabilities is the need for quantitative information, as opposed to qualitative information, when reasoning about non-functional aspects of systems such as throughput, resource utilization, etc. A vast amount of research has been conducted in the area of performance analysis, in which the notion of compositionality typically does not play a major role. In the area of semantics of programming languages and program verification, however, compositionality is a central theme. Various different models with different trade-offs between performance analysis and compositionality have thus been proposed in the literature (see, e.g., [Hil94, Her98, Ber99]). A notion of probabilistic bisimulation that preserves performance metrics is a key ingredient for joint reasoning about qualitative and quantitative behavior, and also for this many proposals have been made.

In earlier work a comparison is made between a number of probabilistic process equivalences (see, e.g., [GSST90, GSS95]) and categorical formulations of Larsen-Skou bisimulation and stochastic bisimulation are given [DEP98, VR99]. In Chapter 2 we focused on the relationship between these and various related notions and made a taxonomy of the most prominent types of probabilistic

bisimulation. In the present chapter we propose a purely coalgebraic perspective on this matter, which allows us to apply a novel general result for the comparison of system types. This way the uniform coalgebraic treatment helps us considerably to clarify the picture and to organize the setting.

As to the comparison of systems, as in Chapter 2, we say that one class of systems is at most as expressive as another if we can map every system of the first type into one of the second such that bisimilarity is *preserved* and *reflected*. For this we require that the transformed system has the same carrier as the original and that two states are bisimilar in the original system if and only if they are bisimilar in the translated one.

The system translations we consider all arise in a straightforward way from natural transformations τ between the two coalgebra functors involved. The translations thus obtained always preserve bisimilarity. The reflection of bisimilarity, however, is not guaranteed in general. For this we present a sufficient condition on the natural transformation τ and the coalgebra functors involved. Interestingly, in our opinion, the result builds on the notion of a *cocongruence* as proposed e.g. by Kurz [Kur00]. This notion is similar to that of a bisimulation, but based on cospans instead of spans—a change of direction which comes in handy for the result we need to prove. We exploit the fact that both notions, bisimulation and cocongruence, characterize the same behavioral equivalence in case the coalgebra functor preserves weak pullbacks.

The expressiveness hierarchy we build with these tools provides a better understanding of the relationship of the various probabilistic system types. The coalgebraic approach facilitated its construction significantly. As far as we know, this form of application of the theory of coalgebras is not reported before in the literature.

The outline of the chapter is as follows: In Section 4.1 we define the different classes of probabilistic systems coalgebraically. We argue that coalgebraic bisimilarity coincides with the standard concrete definitions in Section 4.2. Section 4.3 is the coalgebraic core leading from bisimulation and cocongruences to the result on reflection of bisimilarity. In Section 4.4 we apply the result from Section 4.3 to build the expressiveness hierarchy. We wrap up with conclusions and directions for future work in Section 4.5.

4.1 Probabilistic systems as coalgebras

In this section we model thirteen types of probabilistic systems from the literature on probabilistic modelling as coalgebras. A considerable amount of research has been done on each of these types of systems. They are used as mathematical models of real systems so that formal verification methods based e.g. on temporal logic or process algebra can be applied. Most of the types arose independently in order to improve the modelling of one or another property of a system. One motivating issue is the need to model both non-deterministic and

probabilistic choice. Another issue is the compositional modelling for which operators like hiding (restriction by the environment) and parallel composition play a major role. Therefore some more complex models were proposed that support a definition of these operators. For example, generative systems were extended to bundle probabilistic systems because the former type did not allow for a definition of a natural asynchronous parallel composition operator. In Chapter 2 we gave a wider overview of these models. Here, we just note that the different classes are not defined as coalgebras in the literature. Moreover, in few cases our functorial definition varies from the original one in that we abstract from certain features that are not essential, in our understanding, to the nature of the model under consideration.

We define the systems as coalgebras of suitable behavior functors $\mathcal{F}$. The functors are built using the following syntax

$$\mathcal{F} ::= \mathcal{I}d \mid \underline{A} \mid \mathcal{P} \mid \mathcal{I}d^A \mid \mathcal{D} \mid \mathcal{F} \circ \mathcal{F} \mid \mathcal{F} \times \mathcal{F} \mid \mathcal{F} + \mathcal{F}$$

where the basic functors $\mathcal{I}d, \underline{A}, \mathcal{P}, \mathcal{I}d^A$ and $\mathcal{D}$, as well as the derived functors, were defined in Section 3.5.

In several occasions throughout this chapter we shall need the property of weak pullback preservation. Recall from Section 3.5 that the above syntax produces weak pullback preserving functors.

Again, $\mathsf{Coalg}_{\mathcal{F}}$ denotes the category of coalgebras of the functor $\mathcal{F}$. We fix a set A to serve as a set of actions throughout this chapter.

We now present the probabilistic system types and the functors defining them via Figure 4.1. For each system type the table lists the notation, the functor and the name. For some systems we also include a reference to the bibliographic source of the system. The names used for these systems follow Chapter 2. Some of the names are otherwise not present in the literature. For the Vardi systems sometimes the term *concurrent Markov chains* is used, for the Segala systems the name *(simple) probabilistic automata*, the systems introduced by Pnueli and Zuck are called *probabilistic finite state programs*. We use the name alternating systems following Hansson [Han94], although we do not require strict alternation. Note that the classes of probabilistic systems with strict alternation $\mathbf{SA}$, $\mathbf{SA}_n$ and $\mathbf{SA}_p$ do not find their way in this chapter. The reason is that we could not model them as coalgebras for Set endofunctors, more precisely, the extra condition of strict alternation does not allow for a standard coalgebraic modelling. We introduced the last type of systems ourselves as a generalization of the class $\mathbf{PZ}$ in order to have a top element in the hierarchy.

Basically, every type of probabilistic system arises from the plain definition of a transition system with or without labels. Probabilities can then be added either to every transition, or to transitions labelled with the same action, or there can be a distinction between probabilistic and ordinary (non-deterministic) states, where only the former ones include probabilistic information, or the transition function can be equipped with structure that provides both non-determinism and probability distributions.

$\mathsf{Coalg}_{\mathcal{F}}$	$\mathcal{F}$	name/reference
MC	$\mathcal{D}$	Markov chains
DLTS	$(\mathcal{I}d + 1)^A$	deterministic automata
LTS	$\mathcal{P}(A \times \mathcal{I}d) \cong \mathcal{P}^A$	non-deterministic automata, LTSs
React	$(\mathcal{D} + 1)^A$	reactive systems [LS91, GSST90]
Gen	$\mathcal{D}(A \times \mathcal{I}d) + 1$	generative systems [GSST90]
Str	$\mathcal{D} + (A \times \mathcal{I}d) + 1$	stratified systems [GSST90]
Alt	$\mathcal{D} + \mathcal{P}(A \times \mathcal{I}d)$	alternating systems [Han94]
Var	$(\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$	Vardi systems [Var85]
SSeg	$\mathcal{P}(A \times \mathcal{D})$	simple Segala systems [SL94, Seg95b]
Seg	$\mathcal{PD}(A \times \mathcal{I}d)$	Segala systems [SL94, Seg95b]
Bun	$\mathcal{DP}(A \times \mathcal{I}d)$	bundle systems [DHK98]
PZ	$\mathcal{PDP}(A \times \mathcal{I}d)$	Pnueli-Zuck systems [PZ93]
MG	$\mathcal{PDP}(A \times \mathcal{I}d + \mathcal{I}d)$	most general systems

Figure 4.1: Probabilistic system types

The simplest kind of probabilistic systems that we consider are discrete time, finitely branching Markov chains. Two other classical basic models of probabilistic systems are the reactive and the generative systems. They arise from LTSs when replacing the powerset functor $\mathcal{P}$ by the distribution functor $\mathcal{D}$. At this point we can mention a distinction between systems, the one between *input* type and *output* type of systems. An input system is one defined by a functor of the kind $\mathcal{F}^A$ while an output system has a functor of the form $\mathcal{F}\mathcal{P}(A\times\mathcal{F})$. Note that LTSs can be viewed as both input and output type of systems, due to the isomorphism $\mathcal{P}(A \times \mathcal{I}d) \cong \mathcal{P}^A$. For probabilistic systems this is not the case. As the names already suggest, a reactive system is a probabilistic input system, reacting to the input by the environment, while a generative system is a typical output system, producing output depending on the probability distribution. A reactive system can transit from a given state with a given action to any other state according to the probability distribution that governs this transition. On the other hand in a generative system the distributions involve actions. The generative systems are *fully probabilistic* in the sense that it is enough to erase the action labels on the transitions in order to obtain a Markov chain from a generative system.

Some of the system types introduced above make a distinction between types of states. Such are the stratified systems, the alternating systems, and the Vardi systems. If a state in such a system allows a probabilistic transition, then it is a probabilistic state. If, on the other hand, it allows a (non-)deterministic transition, then it is a (non-)deterministic state. The functor defining the Vardi systems needs more explanation. In a Vardi system $\langle X, \alpha\rangle$, the states can be divided into two sets, a set of non-deterministic states $x \in X$ such that $\alpha(x) \in \mathcal{P}(A \times X)$ and a set of probabilistic states $x \in X$ for which $\alpha(x) \in \mathcal{D}(A \times X)$. The probabilistic states show a generative behavior. Furthermore, by $\bowtie$ we identify some degenerate steps. If from a state $x \in X$ the system can only move, via an action a, to a state $y \in X$, then it is the same as saying that from x, via a, with probability 1 the system moves to y. Therefore, the equivalence $\bowtie$ identifies the Dirac distribution $\mu^1_{\langle a,x\rangle} \in \mathcal{D}(A\times X)$, for $\mu^1_{\langle a,x\rangle}(\langle a, x\rangle) = 1$ and the singleton set $\{\langle a, x\rangle\} \in \mathcal{P}(A \times X)$. This way, there are states in a Vardi system that are both non-deterministic, with one outgoing transition, and probabilistic with a Dirac outgoing transition. By considering $(\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ instead of $\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$, the functorial properties are still preserved.

Unlike reactive and generative ones, systems with the above distinction between states can simulate full non-determinism. Another way of allowing both full non-determinism and probabilities, without distinguishing between states, is by equipping the transition function with a structure, as in the case of Segala, simple Segala, bundle and Pnueli-Zuck systems. The simple Segala model is of input type, enriching the reactive model with full non-determinism, and the other models are of output type, allowing non-determinism in the generative setting.

4.2 Bisimulation correspondence

For most of the probabilistic system types introduced above, a concrete definition of bisimulation is given in the literature and we collected those definitions in Chapter 2. A cornerstone of the coalgebraic approach to bisimulation is the correspondence of bisimilarity of deterministic and non-deterministic transition systems given in concrete terms of transfer conditions [Par81, Mil89] or given in categorial terms of a mediating coalgebra [AM89] (see also [RT93]). De Vink and Rutten have shown [VR99], exploiting the graph-theoretical max-min theorem as in [Jon89], that the concrete notion of bisimulation for Markov chains coincides with the coalgebraic notion. The proof technique extends to most other systems involving the *finite support* probability distribution functor $\mathcal{D}_\omega$ in their definition. As an example, in [BSV03], we sketched the correspondence of concrete bisimulation and coalgebraic bisimulation for the general Segala-type systems (cf. [SL94, Seg95b]) which we modelled there as coalgebras of the functor $\mathcal{PD}_\omega(A \times \mathcal{I}d)$. In [BSV04] we presented another, more modular proof of the correspondence of concrete probabilistic bisimulation with the coalgebraic bisimulation in the case of simple Segala systems. That proof was essentially based on Corollary 3.6.6. At the same time, it was a proof of the correspondence for reactive systems. That technique can also be used in all the other cases.

However, having Section 3.6 available and in particular Section 3.6.2, it is a matter of simple, structured and modular derivation to show the correspondence of coalgebraic and concrete bisimilarity for all of the probabilistic systems that come with a notion of bisimulation. This section is devoted to this correspondence result.

As already mentioned, the bundle probabilistic transition systems [DHK98] do not come equipped with a concrete notion of bisimulation. Equivalence of bundle probabilistic transition systems is defined in terms of the underlying generative probabilistic transitions systems, for which concrete bisimulation coincides with the coalgebraic bisimulation. The approach of Vardi [Var85] and Pnueli and Zuck [PZ93] involves temporal logics. We do not unravel the explicit relationship of logically indistinguishable systems vs. bisimilar ones [LS91]. However, familiarity with coalgebraic bisimulation makes it easy to formulate concrete definitions of bisimulation in the cases of bundle, Vardi and Pnueli-Zuck systems. In Chapter 2 we have given a bisimulation definition for the bundle probabilistic systems and that we will justify here.

Before we present the correspondence results, two remarks are in place:

- A concrete bisimulation is often a relation on the states of one system, while a coalgebraic bisimulation is a relation between the state sets of two systems. We will restrict to coalgebraic bisimulations on the state set of one system and show that two states are related with some coalgebraic bisimulation if and only if they are related with some concrete bisimulation, which gives us the correspondence result. Restriction to the state

set of one system is without loss of generality. It can be shown (provided that $\mathcal{F}$ preserves weak pullbacks) that two states $s \in S$ and $t \in T$ of two $\mathcal{F}$-systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ are related by a bisimulation between S and T if and only if they are related by a bisimulation on the coproduct of the two systems, i.e., $\langle S + T, [\mathcal{F}\iota_1, \mathcal{F}\iota_2] \circ (\alpha + \beta)\rangle$.

- For the correspondence theorem we also need to restrict to coalgebraic bisimulations which are equivalences. This can be done because the coalgebraic bisimilarity is an equivalence for weak-pullback-preserving functors (cf. Section 3.4).

Theorem 4.2.1. *Coalgebraic bisimilarity coincides with concrete bisimilarity, for all types of probabilistic systems from Figure 4.1.*

Proof We will use Lemma 3.6.7 and the results from Section 3.6.2 and provide transfer conditions for bisimulations for each particular type of systems. We will see that if restricted to transfer conditions for equivalence bisimulations on one system, the transfer conditions are equivalent to the ones from the concrete definitions, presented in Chapter 2. This gives us the correspondence results.

The correspondence for Markov chains was already stated in Section 3.6.2. Example 3.6.8 provided us with transfer conditions for labelled transition systems (the class **LTS**). We now derive the transfer conditions for the class of deterministic automata **DLTS**.

A relation $R \subseteq S \times T$ is a bisimulation between the deterministic automata $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if $\langle s, t\rangle \in R$ implies

$$(\forall a \in A)\ \langle \alpha(s)(a), \beta(t)(a)\rangle \ \in\ \mathrm{Rel}(\mathcal{I}d + 1)(R)$$

which, by Lemma 3.6.7(i), (ii) and (3.27) is equivalent to: for all actions a,

$$\begin{array}{ll} \text{either} & \alpha(s)(a) = \beta(t)(a) = \iota_2(*) \\ \text{or} & \langle \alpha(s)(a), \beta(t)(a)\rangle \in R. \end{array}$$

i.e., for all actions a,

$$\begin{array}{l} \text{if } s \xrightarrow{a} s',\ \text{then } t \xrightarrow{a} t' \text{ and } \langle s', t'\rangle \in R,\ \text{and} \\ \text{if } t \xrightarrow{a} t',\ \text{then } s \xrightarrow{a} s' \text{ and } \langle s', t'\rangle \in R. \end{array} \tag{4.1}$$

Hence, we have basically the same transfer condition as for the LTSs (Definition 2.1.14).

Now, since the functor defining the reactive probabilistic systems $(\mathcal{D} + 1)^A$ can be written as $(\mathcal{I}d + 1)^A \circ \mathcal{D}$, by Lemma 3.6.7$(v)$,$(vi)$, and by the previous derivation (4.1), we directly get that the transfer condition for the class **React** is: for all actions a

$$\begin{array}{l} \text{if } s \xrightarrow{a}\!\!\rightsquigarrow \mu,\ \text{then } t \xrightarrow{a}\!\!\rightsquigarrow \nu \text{ and } \mu \equiv_R \nu,\ \text{and} \\ \text{if } t \xrightarrow{a}\!\!\rightsquigarrow \nu,\ \text{then } s \xrightarrow{a}\!\!\rightsquigarrow \mu \text{ and } \mu \equiv_R \nu. \end{array} \tag{4.2}$$

Hence we have obtained exactly the transfer condition from Definition 2.2.3.

Next we derive an auxiliary transfer condition for the functor $\mathcal{D}+1$. The transfer condition is

$$\langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{D}+1)(R)$$

i.e., as in the previous derivations,

$$\begin{array}{l} \text{if } s \rightsquigarrow \mu, \text{ then } t \rightsquigarrow \nu \text{ and } \mu \equiv_R \nu, \text{ and} \\ \text{if } t \rightsquigarrow \nu, \text{ then } s \rightsquigarrow \mu \text{ and } \mu \equiv_R \nu. \end{array} \tag{4.3}$$

For generative systems of type $\mathcal{D} \circ (\underline{A} \times \mathcal{I}d) + 1 = (\mathcal{D}+1) \circ (\underline{A} \times \mathcal{I}d)$, we use again Lemma 3.6.7(vi) and the transfer condition (4.3) for $\mathcal{D}+1$ systems, and we obtain

$$\begin{array}{l} \text{if } s \rightsquigarrow \mu, \text{ then } t \rightsquigarrow \nu \text{ and } \mu \equiv_{\mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)} \nu, \text{ and} \\ \text{if } t \rightsquigarrow \nu \text{ then } s \rightsquigarrow \mu \text{ and } \mu \equiv_{\mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)} \nu. \end{array} \tag{4.4}$$

Moreover, since by Lemma 3.6.7(i), (ii) and (vii):

$$\langle \langle a, s \rangle, \langle b, t \rangle \rangle \in \mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R) \iff a = b \ \wedge \ \langle s, t \rangle \in R$$

it is obvious that

$$\mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R) \ = \ \hat{R} \text{ and } \equiv_{\mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)} = \equiv_{R,A}$$

as in Definition 2.1.6. Hence, the transfer condition for generative systems is exactly the one from Definition 2.2.4.

For the classes **Str** and **Alt**, one can easily get the transfer conditions from Definition 2.2.8. For the simple Segala systems **SSeg**, since $\mathcal{P}(\underline{A} \times \mathcal{D}) = \mathcal{P}(\underline{A} \times \mathcal{I}d) \circ \mathcal{D}$, as in the case of reactive systems, using the transfer condition for LTSs, one gets directly the following transfer condition

$$\begin{array}{l} \text{if } s \stackrel{a}{\rightarrow\!\!\!\!\rightsquigarrow} \mu, \text{ then there exists } \nu \text{ with } t \stackrel{a}{\rightarrow\!\!\!\!\rightsquigarrow} \nu \text{ and } \mu \equiv_R \nu, \text{ and} \\ \text{if } t \stackrel{a}{\rightarrow\!\!\!\!\rightsquigarrow} \nu, \text{ then there exists } \mu \text{ with } s \stackrel{a}{\rightarrow\!\!\!\!\rightsquigarrow} \mu \text{ and } \mu \equiv_R \nu. \end{array} \tag{4.5}$$

Finally, we justify Definition 2.2.18 for bundle probabilistic systems, **Bun**. The functor defining the bundle systems is $\mathcal{D} \circ \mathcal{P}(\underline{A} \times \mathcal{I}d)$. By the transfer condition (3.25) for $\mathcal{D}$-coalgebras, we get that a relation $R \subseteq S \times T$ is a bisimulation between the bundle systems $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ if and only if $\langle s, t \rangle \in R$ implies

$$\text{if } s \rightsquigarrow \mu \text{ and } \ t \rightsquigarrow \nu, \text{ then } \mu \equiv_{\mathrm{Rel}(\mathcal{P}(\underline{A} \times \mathcal{I}d))(R)} \nu. \tag{4.6}$$

We need only to convince ourselves that

$$\equiv_{\mathrm{Rel}(\mathcal{P}(\underline{A} \times \mathcal{I}d))(R)} = \equiv_{R,\mathcal{P}}$$

where $\equiv_{R,\mathcal{P}}$ was defined in Definition 2.2.17. We have

$$\begin{aligned}\mathrm{Rel}(\mathcal{P}(\underline{A} \times \mathcal{I}d))(R) &= \mathrm{Rel}(\mathcal{P})(\mathrm{Rel}(\underline{A} \times \mathcal{I}d)(R)) \\ &= \mathrm{Rel}(\mathcal{P})(\{\langle\langle a, s\rangle, \langle a, t\rangle\rangle \mid \langle s, t\rangle \in R\}) \\ &= \{\langle X, Y\rangle \mid X \subseteq A \times S,\ Y \subseteq A \times T \\ &\qquad (\forall \langle a, s\rangle \in X)(\exists \langle a, t\rangle \in Y)\ \langle s, t\rangle \in R \\ &\qquad (\forall \langle a, t\rangle \in Y)(\exists \langle a, s\rangle \in X)\ \langle s, t\rangle \in R\} \\ &= \equiv_{R,\mathcal{P}}\end{aligned}$$

which completes the proof. □

4.3 Translation of coalgebras

In this section we will prove a technical result about translations of coalgebras. We use translations of $\mathcal{F}$-coalgebras into $\mathcal{G}$-coalgebras in order to compare the expressiveness of coalgebras for different functors, $\mathcal{F}$ and $\mathcal{G}$. Recall that such a translation can easily be obtained from a natural transformation between the two functors under consideration (cf. Definition 3.3.3). Namely, a natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ induces a functor $\mathcal{T}_\tau : \mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$ defined as

$$\mathcal{T}_\tau\langle S, \alpha\rangle = \langle S, \tau_S \circ \alpha\rangle \quad \text{and} \quad \mathcal{T}_\tau h = h.$$

The induced functor is a translation map that always preserves bisimilarity (cf. Section 3.4).

In order to establish that $\mathcal{G}$-coalgebras are at least as expressive as $\mathcal{F}$-coalgebras, we shall use translations $\mathcal{T}_\tau$ for which the converse holds as well, i.e. where s and t are bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle S, \alpha\rangle$ and $\mathcal{T}_\tau\langle T, \beta\rangle$ only if they are bisimilar in the original $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$. In this case we say that $\mathcal{T}_\tau$ *reflects* bisimilarity.

To this end it appears reasonable to ask that the components of $\tau : \mathcal{F} \Longrightarrow \mathcal{G}$ should be injective: Assume that for some set S the component τ_S is not injective, because it identifies two distinct elements $\phi, \psi \in \mathcal{F}S$, i.e. $\tau_S(\phi) = \tau_S(\psi)$. Usually it should not be difficult to find an $\mathcal{F}$-coalgebra structure α on S such that, for two states $s, t \in S$, $\alpha(s) = \phi$ and $\alpha(t) = \psi$ but $s \not\sim t$ in $\langle S, \alpha\rangle$. Since we get $\tau_S(\alpha(s)) = \tau_S(\phi) = \tau_S(\psi) = \tau_S(\alpha(t))$, we have $s \sim t$ in $\mathcal{T}_\tau\langle S, \alpha\rangle = \langle S, \tau_S \circ \alpha\rangle$, which means that $\mathcal{T}_\tau$ does not reflect bisimilarity. (Note though that the above approach does not work in the degenerate case of a functor $\mathcal{F}$ that does not allow non-bisimilar behavior at all, like $\mathcal{F} = \mathcal{I}d$. We shall come back to this example at the end of the section.)

In the following we show that componentwise injectivity of τ implies that $\mathcal{T}_\tau$ reflects a notion of behavioral equivalence defined in terms of *cocongruences* rather than bisimulations. Then we explain that this notion coincides with bisimilarity for coalgebras of functors which preserve weak pullbacks. All coalgebra functors that we consider have this property.

Definition 4.3.1. *A cocongruence between two $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ is a cospan $\langle U, u_1, u_2\rangle$ between S and T, which is jointly surjective, such that there exists an $\mathcal{F}$-coalgebra structure $\gamma : U \to \mathcal{F}U$ making u_1 and u_2 coalgebra homomorphisms, i.e.,*

$$\begin{array}{ccccc} S & \xrightarrow{u_1} & U & \xleftarrow{u_2} & T \\ {\scriptstyle\alpha}\downarrow & & {\scriptstyle\exists\gamma}\downarrow & & \downarrow{\scriptstyle\beta} \\ \mathcal{F}S & \xrightarrow[\mathcal{F}u_1]{} & \mathcal{F}U & \xleftarrow[\mathcal{F}u_2]{} & \mathcal{F}T \end{array}$$

We say that $s \in S$ and $t \in T$ are behavioral equivalent, and write $s \approx t$, in the $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$, if they are identified by some cocongruence between them, i.e., if there exists a cocongruence $\langle U, u_1, u_2\rangle$ with $u_1(s) = u_2(t)$.

We took the name *cocongruence* from Kurz [Kur00, Def. 1.2.1]. Wolter [Wol00] calls these structures *compatible corelations*.

Theorem 4.3.2. *Let $\mathcal{F}$ and $\mathcal{G}$ be two* Set *functors. For a natural transformation $\tau\colon \mathcal{F} \Rightarrow \mathcal{G}$ with injective components we have that $\mathcal{T}_\tau : \mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$ reflects behavioral equivalence.*

For the proof of the theorem we need the following elementary fact.

Lemma 4.3.3. *The category* Set *has the* diagonal fill-in *property for surjective and injective functions: Assume that the outer square in the setting depicted below commutes, where e is surjective and m is injective. Then there exists a unique diagonal arrow d making both of the resulting triangles commute.*

$$\begin{array}{ccc} A & \xrightarrow{e} & B \\ {\scriptstyle f}\downarrow & \overset{\exists! d}{\swarrow} & \downarrow{\scriptstyle g} \\ C & \xrightarrow[m]{} & D \end{array}$$

We proceed with the proof of Theorem 4.3.2.

Proof (of Theorem 4.3.2) Let $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ be two $\mathcal{F}$-coalgebras with states $s \in S$ and $t \in T$ such that $s \approx t$ in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle S, \alpha\rangle$ and $\mathcal{T}_\tau\langle T, \beta\rangle$. So there exists a cocongruence $\langle U, u_1, u_2\rangle$ between the latter two coalgebras identifying s and t. We shall show below that the same cospan is also a cocongruence between the $\mathcal{F}$-coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ that identifies s and t.

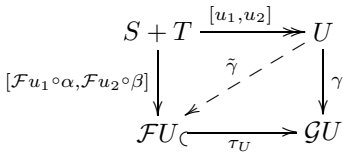
Let $\gamma\colon U \to \mathcal{G}U$ be the transition structure witnessing the cocongruence property of $\langle U, u_1, u_2\rangle$, i.e. both parts of the diagram below commute.

$$\begin{array}{ccccc} S & \xrightarrow{u_1} & U & \xleftarrow{u_2} & T \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\gamma} & & \downarrow{\scriptstyle\beta} \\ \mathcal{F}S & & \mathcal{F}U & & \mathcal{F}T \\ {\scriptstyle\tau_S}\downarrow & & & & \downarrow{\scriptstyle\tau_T} \\ \mathcal{G}S & \xrightarrow[\mathcal{G}u_1]{} & \mathcal{G}U & \xleftarrow[\mathcal{G}u_2]{} & \mathcal{G}T \end{array} \tag{4.7}$$

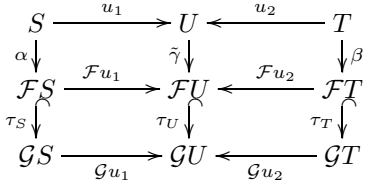
Using this and the naturality of τ in step $(*)$, we compute

$$\begin{aligned}\gamma \circ [u_1, u_2] &= [\gamma \circ u_1, \gamma \circ u_2] \\ &\overset{(4.7)}{=} [\mathcal{G}u_1 \circ \tau_S \circ \alpha, \mathcal{G}u_2 \circ \tau_T \circ \beta] \\ &\overset{(*)}{=} [\tau_U \circ \mathcal{F}u_1 \circ \alpha, \tau_U \circ \mathcal{F}u_2 \circ \beta] \\ &= \tau_U \circ [\mathcal{F}u_1 \circ \alpha, \mathcal{F}u_2 \circ \beta].\end{aligned}$$

This means that the outer square of the diagram below commutes. By the definition of a cocongruence, $[u_1, u_2]$ is surjective and, by assumption, τ_U is injective, so Lemma 4.3.3 provides a diagonal fill-in, say $\tilde{\gamma} : U \to \mathcal{F}U$.



This shows that γ factors as $\tau_U \circ \tilde{\gamma}$, and we can refine picture (4.7) into the one below. It follows from the commutativity of the upper left triangle in the diagram above that the two upper squares in the diagram below indeed commute. So $\tilde{\gamma}$ witnesses that – as wanted – $\langle U, u_1, u_2 \rangle$ is a cocongruence between the original $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$.



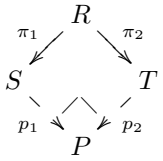
□

We shall show that behavioral equivalence and bisimilarity coincide for coalgebras of a functor that preserves weak pullbacks, so that the above theorem implies that $\mathcal{T}_\tau$ also reflects bisimilarity under appropriate assumptions.

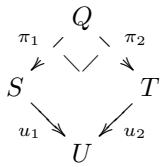
We first demonstrate that we can use pullbacks and pushouts to switch between bisimulations and cocongruences. The argument is standard.

Lemma 4.3.4. *Let $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ be $\mathcal{F}$-coalgebras.*

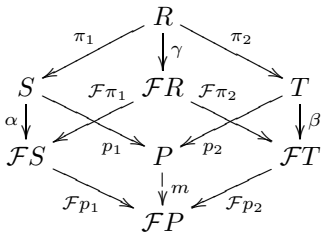
(i) If $R \subseteq S \times T$ is a bisimulation between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ then the pushout $\langle P, p_1, p_2 \rangle$ of R according to the diagram below is a cocongruence between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$.

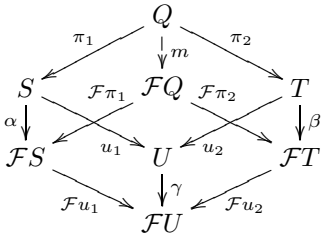


(ii) If $\mathcal{F}$ preserves weak pullbacks and $\langle U, u_1, u_2\rangle$ is a cocongruence between $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ then the pullback $Q = \{\langle s, t\rangle \in S \times T \mid u_1(s) = u_2(t)\}$ of $\langle U, u_1, u_2\rangle$ is a bisimulation between $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$.



Proof (i) Let $\gamma : R \to \mathcal{F}R$ be the coalgebra structure witnessing the bisimulation property. Applying the functor $\mathcal{F}$ to the pushout square we obtain $\mathcal{F}p_1 \circ \mathcal{F}\pi_1 = \mathcal{F}p_2 \circ \mathcal{F}\pi_2$. Together with the bisimulation property this implies that the outer hexagon in the left diagram below commutes. So, by the property of the pushout, there is a unique mediating arrow $m : P \to \mathcal{F}P$ such that $m \circ p_1 = \mathcal{F}p_1 \circ \alpha$ and $m \circ p_2 = \mathcal{F}p_2 \circ \beta$, i.e. $\langle P, p_1, p_2\rangle$ is a cocongruence between $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$.





(ii) Since $\mathcal{F}$ preserves weak pullbacks, $\langle \mathcal{F}Q, \mathcal{F}\pi_1, \mathcal{F}\pi_2\rangle$ is a weak pullback of $\langle \mathcal{F}U, \mathcal{F}u_1, \mathcal{F}u_2\rangle$. Using this and an argument dual to the one for item (i), we get a (not necessarily unique) mediating arrow $m : Q \to \mathcal{F}Q$ in the situation pictured in the right diagram above, which witnesses that Q is a bisimulation between $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$. □

In Section 3.3 we have given a concrete description of pushouts in Set. The following observation about them suffices for our comparison of bisimilarity and behavioral equivalence: the pushout of a relation $R \subseteq S \times T$, i.e. of the span $\langle R, \pi_1, \pi_2\rangle$ identifies all elements related by R. With this we get the following corollary.

Corollary 4.3.5. *Let $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ be two $\mathcal{F}$ coalgebras with states $s \in S$ and $t \in T$.*

(i) If $s \sim t$ then $s \approx t$, i.e. bisimilarity implies behavioral equivalence.

(ii) If $\mathcal{F}$ preserves weak pullbacks, then $s \approx t$ also implies $s \sim t$, i.e. bisimilarity and behavioral equivalence coincide.

Proof If $s \sim t$ then there exists a bisimulation $R \subseteq S \times T$ with $\langle s,t\rangle \in R$. With Lemma 4.3.4 (i) the pushout of R is a cocongruence. Since the pushout identifies all pairs related by R, we get $s \approx t$. For item (ii), let $s \approx t$. This means that there exists a cocongruence $\langle U, u_1, u_2\rangle$ identifying s and t. According to Lemma 4.3.4 (ii), the set of all pairs identified by $\langle U, u_1, u_2\rangle$ is a bisimulation, so $s \sim t$. □

From Theorem 4.3.2 and Corollary 4.3.5 we easily get our result about $\mathcal{T}_\tau$ reflecting bisimilarity.

Theorem 4.3.6. *Let $\tau\colon \mathcal{F} \Rightarrow \mathcal{G}$ be a natural transformation between the* Set*-functors $\mathcal{F}$ and $\mathcal{G}$. If $\mathcal{F}$ preserves weak pullbacks and all components of τ are injective then the functor $\mathcal{T}_\tau$ from Definition 3.3.3 reflects bisimilarity.*

Proof Let $\langle S,\alpha\rangle$ and $\langle T,\beta\rangle$ be $\mathcal{F}$-coalgebras with states $s \in S$ and $t \in T$. If $s \sim t$ in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle S,\alpha\rangle$ and $\mathcal{T}_\tau\langle T,\beta\rangle$ then $s \approx t$ in the same coalgebras according to Corollary 4.3.5 (i). By Theorem 4.3.2 this implies $s \approx t$ in the original $\mathcal{F}$-coalgebras $\langle S,\alpha\rangle$ and $\langle T,\beta\rangle$. Since $\mathcal{F}$ was assumed to preserve weak pullbacks, we can apply Corollary 4.3.5 (ii) to obtain $s \sim t$ in $\langle S,\alpha\rangle$ and $\langle T,\beta\rangle$ as needed. □

The following example demonstrates that Theorem 4.3.6 does not hold without the assumption on weak pullback preservation. It is built on a classical example [AM89] of a functor not preserving weak pullbacks, which is treated in detail also by Gumm and Schröder [GS00].

Example 4.3.7. Consider the functors

$$\mathcal{F}X := \{\langle x,y,z\rangle \in X^3 \mid |\{x,y,z\}| \leq 2\} \quad \text{and} \quad \mathcal{G}X := X^3$$

and the obvious inclusion natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$, all components of which are clearly injective. The functor $\mathcal{F}$ does not preserve weak pullbacks (see [AM89]). To see that the translation $\mathcal{T}_\tau$ does not reflect bisimilarity, consider the $\mathcal{F}$-coalgebra $\langle S,\alpha\rangle$ with

$$S := \{s,t\}, \quad \alpha(s) := \langle s,s,t\rangle, \quad \alpha(t) := \langle s,t,t\rangle.$$

The two states s and t are bisimilar in $\mathcal{T}_\tau\langle S,\alpha\rangle$ but not in $\langle S,\alpha\rangle$. For the first claim, note that $\nabla_S = S \times S$ is a bisimulation on $\mathcal{T}_\tau\langle S,\alpha\rangle$. For the second claim, assume there was a bisimulation $R \subseteq S \times S$ on $\langle S,\alpha\rangle$ with $\langle s,t\rangle \in R$. For the mediating coalgebra structure $\gamma : R \to \mathcal{F}R$ let $\gamma(\langle s,t\rangle) = \langle z_1,z_2,z_3\rangle$. The homomorphism condition implies

$$\langle \pi_1(z_1),\pi_1(z_2),\pi_1(z_3)\rangle = \langle s,s,t\rangle \text{ and } \langle \pi_2(z_1),\pi_2(z_2),\pi_2(z_3)\rangle = \langle s,t,t\rangle.$$

From this we conclude $\gamma(\langle s,t\rangle) = \langle\langle s,s\rangle,\langle s,t\rangle,\langle t,t\rangle\rangle$, but, since all three pairs are different, this is not an element of $\mathcal{F}R$.

The example also suggests that weak pullback preservation is a needed assumption if bisimilarity should relate states with the same observational behavior. Bisimilarity in this case fails to relate s and t, although they cannot be distinguished by external observations.

Coming back to an earlier remark, we mention that componentwise injectivity of the natural transformations τ in Theorem 4.3.6 is not a necessary condition for the reflection of bisimilarity. An example of a natural transformation τ with non-injective components such that $\mathcal{T}_\tau$ still reflects bisimilarity is the natural transformation $! : \mathcal{I}d \Longrightarrow \underline{1}$, with the unique maps $!_S : S \to 1$ into the singleton set $1 = \{*\}$ as components. The translation $\mathcal{T}_!$ trivially reflects bisimilarity, because all states in $\mathcal{I}d$-coalgebras are bisimilar. As it were, the natural transformation forgets only information that is not relevant for bisimilarity. Another, more interesting example of the same kind is the natural transformation that maps probability distributions on their set of support. However, we are not aware of any examples involving a functor $\mathcal{F}$ such that there are $\mathcal{F}$-coalgebras with non-bisimilar states.

4.4 The hierarchy

We will exploit Theorem 4.3.6 to achieve the primary goal of this chapter, viz. establishing a hierarchy of probabilistic system types. The hierarchy is the one already presented in Chapter 2, except for the strictly alternating classes. Theorem 4.3.6 and the coalgebraic approach in general lead to a brief and elegant proof of the hierarchy result.

Let $\mathcal{F}$ and $\mathcal{G}$ be functors on Set. If there exists a translation functor from $\mathsf{Coalg}_\mathcal{F}$ to $\mathsf{Coalg}_\mathcal{G}$ that both preserves and reflects bisimilarity then we say that the class $\mathsf{Coalg}_\mathcal{F}$ is *coalgebraically embedded* in the class $\mathsf{Coalg}_\mathcal{G}$. This relation is clearly reflexive and transitive.

The expressiveness criterion makes sure that if a class of systems $\mathbf{A}$ is coalgebraically embedded in a class $\mathbf{B}$ then a "copy" of any system belonging to $\mathbf{A}$ exists in $\mathbf{B}$, and therefore we consider the class $\mathbf{B}$ at least as expressive as the class $\mathbf{A}$. Another hierarchy result, using a different expressiveness criterion is given for the reactive, generative and stratified systems by Van Glabbeek et al. [GSST90, GSS95]. According to the expressiveness criterion of Van Glabbeek et al. the class $\mathbf{A}$ is at least as expressive as the class $\mathbf{B}$ if there exists a translation functor from $\mathbf{A}$ to $\mathbf{B}$ that preserves bisimilarity. Their expressiveness criterion is local: any system of $\mathbf{A}$ can be considered as expressing at least as much as its image in $\mathbf{B}$, while our expressiveness criterion is global: each system in $\mathbf{A}$ expresses exactly the same as its image, but the class $\mathbf{B}$ may be "bigger".

The next theorem lists coalgebraic embeddings between the probabilistic system types introduced in Figure 4.1.

Theorem 4.4.1. *The coalgebraic embeddings presented in Figure 4.2 hold*

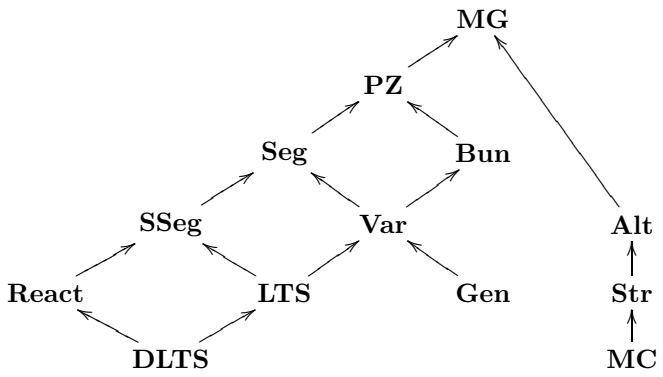


Figure 4.2: Hierarchy of probabilistic system types

among the probabilistic system types, where an arrow $\mathbf{A} \to \mathbf{B}$ *expresses that the class* $\mathbf{A}$ *is coalgebraically embeddable in the class* $\mathbf{B}$*.*

Proof By Theorem 4.3.6, if $\mathcal{F}, \mathcal{G}$ are functors on Set such that $\mathcal{F}$ preserves weak pullbacks and there is a componentwise injective natural transformation from $\mathcal{F}$ to $\mathcal{G}$, then $\mathsf{Coalg}_{\mathcal{F}}$ is coalgebraically embeddable in $\mathsf{Coalg}_{\mathcal{G}}$.

Having the weak pullback preservation for all functors from Figure 4.1, it is enough to construct a componentwise injective natural transformation for each embedding. We start by defining some elementary natural transformations and collecting some simple properties. Let $\mathcal{F}, \mathcal{G}, \mathcal{H}$ be functors on Set.

- We define the *empty* natural transformation $1 \stackrel{\eta}{\Longrightarrow} \mathcal{P}$, for $\eta_X(*) = \emptyset$.
- The left and right coproduct injections ι_1 and ι_2 are natural transformations $\mathcal{F} \stackrel{\iota_1}{\Longrightarrow} \mathcal{F} + \mathcal{G}$, $\mathcal{G} \stackrel{\iota_2}{\Longrightarrow} \mathcal{F} + \mathcal{G}$ with injective components.
- For every set X, the injective functions $\sigma_X : X \to \mathcal{P}X$ where $\sigma_X(x) = \{x\}$ form a natural transformation $\mathcal{I}d \stackrel{\sigma}{\Longrightarrow} \mathcal{P}$, the *singleton* natural transformation.
- For every set X, the injective functions $\delta_X : X \to \mathcal{D}X$ where $\delta_X(x) = \mu^1_x$, $\mu^1_x(x) = 1$ form the *Dirac* natural transformation $\mathcal{I}d \stackrel{\delta}{\Longrightarrow} \mathcal{D}$.
- For any set X, the injective functions $\phi_X : (X+1)^A \to \mathcal{P}(A \times X)$ defined by $\phi_X(f) = Graph(f) = \{\langle a, f(a)\rangle \mid f(a) \in X\}$ for $f : A \to X + 1$, form a natural transformation $(\mathcal{I}d + 1)^A \stackrel{\phi}{\Longrightarrow} \mathcal{P}(A \times \mathcal{I}d)$
- From $\mathcal{F} \stackrel{\tau_1}{\Longrightarrow} \mathcal{H}$ and $\mathcal{G} \stackrel{\tau_2}{\Longrightarrow} \mathcal{H}$ we get a natural transformation $\mathcal{F}+\mathcal{G} \stackrel{[\tau_1,\tau_2]}{\Longrightarrow} \mathcal{H}$.

- If $\mathcal{F}_1 \overset{\tau_1}{\Longrightarrow} \mathcal{G}_1$ and $\mathcal{F}_2 \overset{\tau_2}{\Longrightarrow} \mathcal{G}_2$ are componentwise injective, then so is the natural transformation $\mathcal{F}_1 + \mathcal{F}_2 \overset{\tau_1+\tau_2}{\Longrightarrow} \mathcal{G}_1 + \mathcal{G}_2$.

- If $\mathcal{F} \overset{\tau}{\Longrightarrow} \mathcal{G}$ is componentwise injective, then so is $\mathcal{F}\mathcal{H} \overset{\tau\mathcal{H}}{\Longrightarrow} \mathcal{G}\mathcal{H}$, where $(\tau\mathcal{H})_X = \tau_{\mathcal{H}X}$.

- From $\mathcal{F} \overset{\tau}{\Longrightarrow} \mathcal{G}$ we get a natural transformation $\mathcal{H}\mathcal{F} \overset{\mathcal{H}\tau}{\Longrightarrow} \mathcal{H}\mathcal{G}$ with $(\mathcal{H}\tau)_X = \mathcal{H}(\tau_X)$. If the functor $\mathcal{H}$ preserves injectivity and all components of τ are injective, then so are the components of $\mathcal{H}\tau$. For the first condition, since every Set-functor preserves injectives with nonempty domain, we just need to check that $\mathcal{H}$ maps functions from the empty set to injective functions. This is the case for $\mathcal{P}$, $\mathcal{D}$, and the other functors we use below, as one directly verifies.

Now we prove all the coalgebraic embeddings, by building the needed natural transformations from the elementary ones mentioned above.

MC → Str: $\mathcal{D} \overset{\iota_1}{\Longrightarrow} \mathcal{D} + (A \times \mathcal{I}d) + 1$

DLTS → LTS: $(\mathcal{I}d + 1)^A \overset{\phi}{\Longrightarrow} \mathcal{P}(A \times \mathcal{I}d)$

DLTS → React: $(\mathcal{I}d + 1)^A \overset{\mathcal{F}\delta}{\Longrightarrow} (\mathcal{D} + 1)^A$, for $\mathcal{F} = (\mathcal{I}d + 1)^A$.

React → SSeg: $(\mathcal{D} + 1)^A \overset{\phi\mathcal{D}}{\Longrightarrow} \mathcal{P}(A \times \mathcal{D})$

LTS → SSeg: $\mathcal{P}(A \times \mathcal{I}d) \overset{\mathcal{F}\delta}{\Longrightarrow} \mathcal{P}(A \times \mathcal{D})$, for $\mathcal{F} = \mathcal{P}(A \times \mathcal{I}d)$.

LTS → Var: $\mathcal{P}(A \times \mathcal{I}d) \overset{\xi\circ\iota_2}{\Longrightarrow} (\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ for $\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d) \overset{\xi}{\Longrightarrow} (\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ being the canonical natural transformation, that maps every element to its class. Although ξ is not injective, $\xi \circ \iota_2$ is.

Gen → Var: $\mathcal{D}(A \times \mathcal{I}d) + 1 \overset{\xi\circ(id+\eta\mathcal{F})}{\Longrightarrow} (\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$, for $\mathcal{F} = A \times \mathcal{I}d$. The transformation $\xi \circ (id + \eta\mathcal{F})$ is componentwise injective, since $id + \eta\mathcal{F}$ does not reach $\bowtie$-identifiable elements in $\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$.

Var → Seg: $(\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie \overset{[\sigma\mathcal{D},\mathcal{P}\delta]\mathcal{F}}{\Longrightarrow} \mathcal{P}\mathcal{D}(A \times \mathcal{I}d)$ for $\mathcal{F} = A \times \mathcal{I}d$. Note that the natural transformation factors through the equivalence classes, because the $\bowtie$-identified elements are mapped to the same Segala behavior. The transformation is injective.

Var → Bun: $(\mathcal{D}(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie \overset{[\mathcal{D}\sigma,\delta\mathcal{P}]\mathcal{F}}{\Longrightarrow} \mathcal{D}\mathcal{P}(A \times \mathcal{I}d)$ for $\mathcal{F} = A \times \mathcal{I}d$. As in the case **Var → Seg**, the $\bowtie$-identified elements are mapped to the same bundle behavior, and the transformation is injective.

SSeg → Seg: $\mathcal{P}(A \times \mathcal{D}) \stackrel{\mathcal{P}\tau}{\Longrightarrow} \mathcal{P}\mathcal{D}(A \times \mathcal{I}d)$ where $(A \times \mathcal{D}) \stackrel{\tau}{\Longrightarrow} \mathcal{D}(A \times \mathcal{I}d)$ is given by $\tau_X(\langle a, \mu\rangle) = \mu_a^1 \times \mu$, where $\mu \times \mu'(\langle x, x'\rangle) = \mu(x) \cdot \mu'(x')$ and μ_a^1 is the Dirac distribution for a. All components of τ are injective.

Str → Alt: $\mathcal{D} + (A \times \mathcal{I}d) + 1 \stackrel{id+[\sigma,\eta]\mathcal{F}}{\Longrightarrow} \mathcal{D} + \mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = A \times \mathcal{I}d$. Componentwise injectivity holds.

Seg → PZ: $\mathcal{P}\mathcal{D}(A \times \mathcal{I}d) \stackrel{\mathcal{P}\mathcal{D}\sigma\mathcal{F}}{\Longrightarrow} \mathcal{P}\mathcal{D}\mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = A \times \mathcal{I}d$.

Bun → PZ: $\mathcal{D}\mathcal{P}(A \times \mathcal{I}d) \stackrel{\sigma\mathcal{F}}{\Longrightarrow} \mathcal{P}\mathcal{D}\mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = \mathcal{D}\mathcal{P}(A \times \mathcal{I}d)$.

PZ → MG: $\mathcal{P}\mathcal{D}\mathcal{P}(A \times \mathcal{I}d) \stackrel{\mathcal{P}\mathcal{D}\mathcal{P}\iota_1}{\Longrightarrow} \mathcal{P}\mathcal{D}\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$

Alt → MG: $\mathcal{D} + \mathcal{P}(A \times \mathcal{I}d) \stackrel{\sigma\mathcal{H} \circ [\mathcal{D}(\sigma\mathcal{F} \circ \iota_2), \delta\mathcal{G} \circ \mathcal{P}\iota_1]}{\Longrightarrow} \mathcal{P}\mathcal{D}\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$. Here injections go to $A \times \mathcal{I}d + \mathcal{I}d$ and $\mathcal{F} = A \times \mathcal{I}d + \mathcal{I}d$, $\mathcal{G} = \mathcal{P}\mathcal{F}$, $\mathcal{H} = \mathcal{D}\mathcal{G} = \mathcal{D}\mathcal{P}\mathcal{F}$. Again, there is no overlap between the images in the two cases.

□

We note here that some more arrows than those presented in Figure 4.2 may exist. Our results do not provide a way of showing absence of arrows. For instance in case of a countable label set A, we get **React** → **Gen** by the transformation $\tau : (\mathcal{D} + 1)^A \Rightarrow \mathcal{D}(A \times \mathcal{I}d) + 1$ defined in the following way. Fix a distribution $\mu \in \mathcal{D}A$ such that $spt(\mu) = A$. For any set X and any $\phi : A \to \mathcal{D}X + 1$, define $\tau_X(\phi) = *$ if and only if $\phi(a) = *$ for all $a \in A$ and otherwise, $\tau_X(\phi) = \nu \in \mathcal{D}(A \times \mathcal{I}d)$ where for $a \in A, x \in X$

$$\nu(a, x) = \begin{cases} 0 & \text{if } \phi(a) = *, \\ \frac{\phi(a)(x) \cdot \mu(a)}{\mu[\{b \in A \mid \phi(b) \neq *\}]} & \text{otherwise.} \end{cases}$$

The transformation τ is natural and its components are injective. However, we can argue that this transformation can not be defined for arbitrary set A.

4.5 Conclusions and future work

We have studied a relation between the classes of coalgebras of several Set-functors that arise naturally from the literature on probabilistic and nondeterministic systems. We proved a general embeddability result and used it to establish a hierarchy of probabilistic system types. The hierarchy pictures the expressive power of system behavior types that differ mainly in the combination of nondeterminism and probability.

However, we lack a general way to prove that one class is strictly more expressive than another. A deeper study of expressiveness should try to find the boundaries by also establishing negative embeddability results. We leave this task for future work. Some alternative characterization of what it means that one class of

systems is embeddable in another may be helpful here. Another direction for further research is a similar classification of essentially continuous systems, in addition to the discrete systems that we have focused on so far.

5

Weak bisimulation

We propose a coalgebraic definition of weak bisimulation for classes of coalgebras obtained from bifunctors in the category Set. Weak bisimilarity for a system is obtained as strong bisimilarity of a transformed system. The transformation consists of two steps: First, the behavior on actions is expanded to behavior on finite words. Second, the behavior on finite words is taken modulo the hiding of invisible actions, yielding behavior on equivalence classes of words closed for silent steps. The coalgebraic definition is justified by two correspondence results: one for the classical notion of weak bisimulation of Milner, another for the notion of weak bisimulation for generative probabilistic transition systems as advocated by Baier and Hermanns.

In this chapter we present a definition of weak bisimulation for action type systems. A typical example of an action type system is the familiar labelled transition system (LTS) (see, e.g., [Plo81, Mil90]), but also many types of probabilistic systems (see, e.g., [LS91, SL94, GSS95, BH97, Seg95b]) fall into this class. In order to emphasize the role of the actions we view coalgebras as arising from bifunctors over Set.

For the verification of properties of a system strong bisimilarity is often too strong an equivalence. Weak bisimilarity [Mil80, Mil90] is a looser equivalence on systems that abstracts away from invisible steps; weak bisimilarity for a labelled transition system $\mathcal{S}$ amounts to strong bisimilarity on the 'double-arrowed' system $\mathcal{S}'$ induced by $\mathcal{S}$. We exploit this idea for a general coalgebraic definition of weak bisimulation. Our approach, given a system $\mathcal{S}$, consists of two stages.

1. First, we define a '$*$-extension' $\mathcal{S}'$ of $\mathcal{S}$ which is a system with the same carrier as $\mathcal{S}$, but with action set A^*, the set of all finite words over A. The system $\mathcal{S}'$ captures the behavior of $\mathcal{S}$ on finite traces.
2. Next, given a set of invisible actions $\tau \subseteq A$, we transform $\mathcal{S}'$ into a 'weak-τ-extension' $\mathcal{S}''$ which abstracts away from τ steps. Then we define weak bisimilarity on $\mathcal{S}$ as strong bisimilarity on the weak-τ-extension $\mathcal{S}''$.

For LTS weak bisimulation is an established notion. In the context of concrete

probabilistic transition systems, there have been several proposals for a notion of weak bisimulation, often relying on the particular model under consideration. Segala [SL94, Seg95b] proposed four notions of weak relations for his model of simple probabilistic automata. Baier and Hermanns [BH97, Bai98, BH99] have given a rather appealing definition of weak bisimulation for generative probabilistic systems. Philippou, Lee and Sokolsky [PLS00] studied weak bisimulation in the setting of the alternating model [Han91]. This work was extended to infinite systems by Desharnais, Gupta, Jagadeesan and Panangaden [DGJP02b]. The same authors also provided a metric analogue of weak bisimulation [DGJP02a].

Here, we work in a coalgebraic framework and use the general coalgebraic apparatus of bisimulation [AM89, JR96, Rut00]. For weak bisimulation in this setting, there has been early work by Rutten on weak bisimulation for while programs [Rut99] succeeded by a syntactic approach to weak bisimulation by Rothe [Rot02]. In the latter paper, weak bisimulation for a particular class of coalgebras was obtained by transforming a coalgebra into an LTS and making use of Milner's weak bisimulation there. This approach also enabled a definition of weak homomorphisms and weak simulation relations. Later, in the work of Rothe and Mašulović [RM02] a complex, but interesting coalgebraic theory was developed leading to weak bisimulation for functors that weakly preserve pullbacks. They also consider a chosen 'observer' and hidden parts of a functor. However, in the case of probabilistic and similar systems, it does not lead to intuitive results and can not be related to the concrete notions of weak bisimulation mentioned above. The so-called skip relations used in [RM02] seem to be the major obstacle as it remains unclear how quantitative information can be incorporated.

The two-phase approach of defining weak bisimilarity is, amplifying Milner's original idea, rather natural. In the category theoretical setting it has been suggested in the context of the open map treatment of weak bisimulation on presheaf models [FCW99]. Our proposal builds up on the intuition from concrete cases. We focus only on hiding actions and we provide a (parameterized) definition of weak bisimulation for action-type coalgebras. A drawback of our approach is that the definition of weak bisimulation is parameterized with a notion of a $*$-extension that does not come from a general categorical construction but has to be defined ad-hoc for concrete types of systems. We are able to prove, not only for the case of labelled transition systems, but also for probabilistic systems that our coalgebraic proposal corresponds to the concrete definitions of [Mil90] and [BH97]. Despite the appeal of the coalgebraic definition of weak bisimulation, proofs of correspondence results may vary from straightforward to technically involved. For example, the relevant theorem for labelled transition systems takes less than a page, whereas proving the correspondence result for generative probabilistic systems takes in its present form more than twenty pages (additional machinery included).

The chapter is organized as follows: Section 5.1 presents the definition of weak

bisimulation. We show that our definition of weak bisimilarity leads to Milner's weak bisimilarity for LTSs in Section 5.2. Section 5.3 is devoted to a correspondence result for the class of generative systems of the notion of weak bisimilarity of Baier and Hermanns and our coalgebraic definition. Finally, Section 5.4 draws some conclusions.

5.1 Weak bisimulation for action-type coalgebras

In this section we present a general definition of weak bisimulation for action-type systems. Our idea arises as a generalization of what is known from the literature for concrete types of systems. In our opinion, a weak bisimulation on a given system must be a strong bisimulation on a suitably transformed system obtained from the original one.

Weak bisimulation in concrete cases deals with hiding invisible actions. Therefore we focus on weak bisimulation for systems that can perform some actions. Such systems are the action-type coalgebras. Recall that we have defined action-type coalgebras in Definition 3.2.4 as triples $\langle S, A, \alpha\rangle$ such that $\langle S, \alpha : S \to \mathcal{F}_A S\rangle$ is a coalgebra for the functor $\mathcal{F}_A$ induced by a bifunctor $\mathcal{F}$, as in Equation (3.1). Specifying the set A in the definition of action-type coalgebras emphasizes the set of possible actions. We denote by $\mathsf{Coalg}_{\mathcal{F}}^{A}$ the category of action-type coalgebras defined by a bifunctor $\mathcal{F}$ with action set A.

Before we discuss weak bisimilarity in general, we fix two examples that we will consider in detail in this chapter: LTSs and generative probabilistic systems. We note that LTSs are action-type coalgebras for the functor $\mathcal{L}_A$, derived from the bifunctor

$$\mathcal{L} = \mathcal{P}(\mathcal{I}d \times \mathcal{I}d).$$

The generative probabilistic systems are also action-type coalgebras of type $\mathcal{G}_A$ corresponding to the bifunctor

$$\mathcal{G} = \mathcal{D}(\mathcal{I}d \times \mathcal{I}d) + 1.$$

We proceed with the definition of weak bisimulation for action-type coalgebras. The definition consists of two phases. First we define a $*$-*extended system*, that captures the behavior of the original system when extending from the given set of actions A to A^*, the set of finite words over A. The $*$-extension should emerge from the original system in a faithful way (which will be made precise below). The second phase considers invisibility. Given a subset $\tau \subseteq A$ of invisible actions, we restrict the $*$-extension to visible behavior only, by defining a so-called, *weak-τ-extended system*. Then a weak bisimulation relation on the original system is any bisimulation relation on the weak-τ-extension.

Definition 5.1.1. *Let $\mathcal{F}$ and $\mathcal{G}$ be two bifunctors. Let Φ be a map assigning to every $\mathcal{F}_A$ coalgebra $\langle S, A, \alpha\rangle$, a $\mathcal{G}_{A^*}$ system $\langle S, A^*, \alpha'\rangle$, on the same state set, such that the following conditions are met*

(1) Φ *is injective, i.e.* $\Phi(\langle S, A, \alpha\rangle) = \Phi(\langle S, A, \beta\rangle) \Rightarrow \alpha = \beta$*;*

(2) Φ *preserves and reflects bisimilarity, i.e.* $s \sim t$ *in the system* $\langle S, A, \alpha\rangle$ *if and only if* $s \sim t$ *in the transformed system* $\Phi(\langle S, A, \alpha\rangle)$*.*

Then Φ *is called a* $*$*-translation, notation* $\Phi : \mathcal{F} \stackrel{*}{\to} \mathcal{G}$*, and we say that* $\Phi(\langle S, A, \alpha\rangle)$ *is a* $*$*-extension of* $\langle S, A, \alpha\rangle$*.*

The conditions (1) and (2) in Definition 5.1.1 make sure that the original system is "embedded" in its $*$-extension, cf. Chapter 2 and Chapter 4. The fact that a $*$-translation may lead to systems of a new type, viz. of the bifunctor $\mathcal{G}$, might seem counterintuitive at first sight. However, this extra freedom is exploited in Section 5.3 when the starting functor is not expressive enough to allow for a $*$-extension of generative systems.

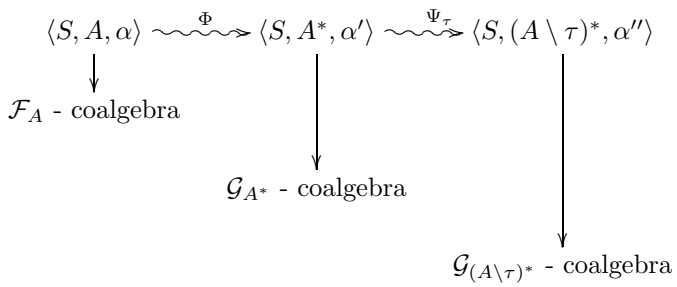
A way to obtain $*$-translations follows from a previous result. Namely, if $\lambda\colon \mathcal{F}_A \Rightarrow \mathcal{G}_{A^*}$ is a natural transformation with injective components and the functor $\mathcal{F}_A$ preserves weak pullbacks, then the induced functor (see Definition 3.3.3) is a $*$-translation, according to Theorem 4.3.6. However, we shall see later that $*$-translations emerging from natural transformations do not cover known concrete cases.

Having extended an $\mathcal{F}_A$ system to its $*$-extension we show how to hide invisible actions. Let $\tau \subseteq A$. Consider the function $h_\tau\colon A^* \to (A \setminus \tau)^*$ induced by: $h_\tau(a) = a$ if $a \notin \tau$ and $h_\tau(a) = \varepsilon$ for $a \in \tau$ (where ε denotes the empty word). The function h_τ is deleting all the occurrences of elements of τ in a word of A^*. Denote by A_τ the set $A_\tau = (A \setminus \tau)^*$. By Lemma 3.3.2, we get the following.

Corollary 5.1.2. *The transformation* $\eta^\tau\colon \mathcal{G}_{A^*} \Rightarrow \mathcal{G}_{A_\tau}$ *given by* $\eta^\tau_S = \mathcal{G}\langle h_\tau, id_S\rangle$ *is natural.* □

Let Ψ_τ be the functor from $\mathsf{Coalg}_{\mathcal{G}}^{A^*}$ to $\mathsf{Coalg}_{\mathcal{G}}^{A_\tau}$ induced by the natural transformation η^τ, i.e. $\Psi_\tau(\langle S, A^*, \alpha'\rangle) = \langle S, A_\tau, \alpha''\rangle$ for $\alpha'' = \eta^\tau_S \circ \alpha'$ and $\Psi_\tau f = f$ for any morphism $f : S \to T$ (see Definition 3.3.3). As mentioned above, the induced functor preserves bisimilarity. The composition of a $*$-translation Φ and the hiding functor Ψ_τ is denoted by $W_\tau = \Psi_\tau \circ \Phi$ and is called a weak-τ-translation. The resulting system is the weak-τ-extension of $\langle S, A, \alpha\rangle$.

The transformation to a weak-τ-extension is presented in the following scheme.



A weak-τ-translation, or equivalently, the pair $\langle \Phi, \tau\rangle$, yields a notion of weak bisimulation with respect to Φ and τ.

Definition 5.1.3. *Let $\mathcal{F}$, $\mathcal{G}$ be two bifunctors, $\Phi : \mathcal{F} \xrightarrow{*} \mathcal{G}$ a $*$-translation and $\tau \subseteq A$. Let $\langle S, A, \alpha\rangle$ and $\langle T, A, \beta\rangle$ be two $\mathcal{F}_A$ systems. A relation $R \subseteq S \times T$ is a weak bisimulation w.r.t. $\langle \Phi, \tau\rangle$ if and only if it is a bisimulation between $W_\tau(\langle S, A, \alpha\rangle)$ and $W_\tau(\langle T, A, \beta\rangle)$. Two states $s \in S$ and $t \in T$ are weakly bisimilar w.r.t. $\langle \Phi, \tau\rangle$, notation $s \approx_\tau t$, if they are related by some weak bisimulation w.r.t. $\langle \Phi, \tau\rangle$.*

Next we prove that any relation $\approx_\tau$ obtained in this way, satisfies the properties that are intuitively expected from a weak bisimilarity relation.

Lemma 5.1.4. *Let $\mathcal{F}$, $\mathcal{G}$ be two bifunctors, $\Phi : \mathcal{F} \xrightarrow{*} \mathcal{G}$ a $*$-translation, $\langle S, A, \alpha\rangle$ an $\mathcal{F}_A$-coalgebra, $\tau \subseteq A$ and let $\approx_\tau$ denote the weak bisimilarity on $\langle S, A, \alpha\rangle$ w.r.t. $\langle \Phi, \tau\rangle$. Then the following hold:*

(*i*) $\sim \subseteq \approx_\tau$ *for any* $\tau \subseteq A$
i.e. strong bisimilarity implies weak.

(*ii*) $\sim = \approx_\emptyset$
i.e. strong bisimilarity is weak bisimilarity in absence of invisible actions.

(*iii*) $\tau_1 \subseteq \tau_2 \Rightarrow \approx_{\tau_1} \subseteq \approx_{\tau_2}$ *for any* $\tau_1, \tau_2 \subseteq A$.
i.e. the more actions are invisible, the coarser the weak bisimilarity gets.

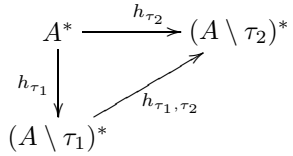
Proof

(*i*) Assume $s \sim t$ in $\langle S, A, \alpha\rangle$. Since Φ preserves bisimilarity (Definition 5.1.1) we have that $s \sim t$ in $\Phi(\langle S, A, \alpha\rangle)$. Next, since Ψ_τ preserves bisimilarity we get $s \sim t$ in $\Psi_\tau \circ \Phi(\langle S, A, \alpha\rangle)$, which by Definition 5.1.3 means $s \approx_\tau t$ in $\langle S, A, \alpha\rangle$.

(*ii*) From (*i*) we get $\sim \subseteq \approx_\emptyset$. For the opposite inclusion, note that the natural transformation $\eta^\emptyset$ from Corollary 5.1.2 is the identity natural transformation. Therefore the induced functor $\Psi_\emptyset$ is the identity functor on $\mathsf{Coalg}_{\mathcal{G}}^{A^*}$. Now assume $s \approx_\emptyset t$ in $\langle S, A, \alpha\rangle$. This means $s \sim t$ in $W_\emptyset(\langle S, A, \alpha\rangle)$,

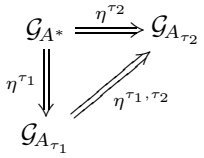
i.e. $s \sim t$ in $\Psi_\emptyset \circ \Phi(\langle S, A, \alpha\rangle)$, i.e. $s \sim t$ in $\Phi(\langle S, A, \alpha\rangle)$. Since, by Definition 5.1.1, every $*$-translation reflects bisimilarity we get $s \sim t$ in $\langle S, A, \alpha\rangle$.

(iii) Let $\tau_1 \subseteq \tau_2$. Consider the diagram



where h_{τ_1,τ_2} is the map deleting all occurrences of elements of τ_2 in a word of $(A \setminus \tau_1)^*$. The diagram commutes since first deleting all occurrences of elements of τ_1 followed by deleting all occurrences of elements of τ_2, in a word of A^* is the same as just deleting all occurrences of elements of τ_2.

Denote by η^{τ_1}, η^{τ_2}, η^{τ_1,τ_2} the natural transformations from Corollary 5.1.2, Lemma 3.3.2, corresponding to h_{τ_1}, h_{τ_2}, h_{τ_1,τ_2} respectively. They make the following diagram commute.



Since the functors Ψ_{τ_1}, Ψ_{τ_2}, Ψ_{τ_1,τ_2} are induced by the natural transformations η^{τ_1}, η^{τ_2}, η^{τ_1,τ_2}, respectively, by Definition 3.3.3 it holds that

$$\Psi_{\tau_2} = \Psi_{\tau_1,\tau_2} \circ \Psi_{\tau_1} \tag{5.1}$$

and they all preserve bisimilarity. Now assume $s \approx_{\tau_1} t$ in $\langle S, A, \alpha\rangle$. This means that $s \sim t$ in the system $\Psi_{\tau_1} \circ \Phi(\langle S, A, \alpha\rangle)$. Then, since Ψ_{τ_1,τ_2} preserves bisimilarity we have $s \sim t$ in the system $\Psi_{\tau_1,\tau_2} \circ \Psi_{\tau_1} \circ \Phi(\langle S, A, \alpha\rangle)$ which by equation (5.1) is the system $\Psi_{\tau_2} \circ \Phi(\langle S, A, \alpha\rangle)$ and we find $s \approx_{\tau_2} t$ in $\langle S, A, \alpha\rangle$.

□

For further use, we introduce some more notation. For any $w \in A_\tau$, we denote $B_w = h_\tau^{-1}(\{w\}) \subseteq A^*$. We refer to the sets B_w as blocks. Note that $B_w = \tau^* a_1 \tau^* \cdots \tau^* a_k \tau^*$ for $w = a_1 \ldots a_k \in A_\tau = (A \setminus \tau)^*$.

5.2 Weak bisimulation for LTSs

In this section we show that in the case of LTS there exists a $*$-translation according to the general definition, such that weak bisimulation in the concrete

case [Mil89] coincides with weak bisimulation induced by this $*$-translation. First we recall the definition of concrete weak bisimulation for LTSs.

Definition 5.2.1. *Let $\langle S, A, \alpha\rangle$ be an LTS. Assume $\tau \in A$ is an invisible action. An equivalence relation $R \subseteq S \times S$ is a weak bisimulation on $\langle S, A, \alpha\rangle$ if and only if $\langle s, t\rangle \in R$ implies that*

$$\text{if } s \xrightarrow{a} s', \text{ then there exists } t' \in S \text{ with } t \xrightarrow{\tau}{}^* \circ \xrightarrow{a} \circ \xrightarrow{\tau}{}^* t' \text{ and } \langle s', t'\rangle \in R$$

for all $a \in A \setminus \{\tau\}$, and

$$\text{if } s \xrightarrow{\tau} s', \text{ then there exists } t' \in S \text{ with } t \xrightarrow{\tau}{}^* t' \text{ and } \langle s', t'\rangle \in R.$$

Two states s and t are called weakly bisimilar if and only if they are related by some weak bisimulation relation. Notation $s \approx_l t$.

We now present a definition of a $*$-translation that will give us the same weak bisimilarity relation. Let $\mathcal{L}$, $\mathcal{L}_A$ be the functors for LTSs, as introduced in Section 5.1.

Definition 5.2.2. *Let Φ assign to every LTS, i.e. any $\mathcal{L}_A$ coalgebra $\langle S, A, \alpha\rangle$ the $\mathcal{L}_{A^*}$ coalgebra $\langle S, A^*, \alpha'\rangle$ where for $w = a_1 \dots a_k \in A^*$,*

$$\langle a_1 \dots a_k, s'\rangle \in \alpha'(s) \quad \Longleftrightarrow \quad s \xrightarrow{a_1} \circ \xrightarrow{a_2} \circ \cdots \circ \xrightarrow{a_k} s'.$$

We use the notation $s \overset{w}{\Rightarrow} s'$ for $\langle w, s'\rangle \in \alpha'(s)$.

Hence, for $w = a_1 \dots a_k$, we have $s \overset{w}{\Rightarrow} s'$ if and only if there exist states $s_1, \dots, s_{k-1}$ such that

$$s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_{k-1}} s_{k-1} \xrightarrow{a_k} s_k.$$

Furthermore, note that for $a \in A$, since no hiding applies, it holds that

$$s \xrightarrow{a} s' \text{ in } \langle S, A, \alpha\rangle \quad \text{if and only if} \quad s \overset{a}{\Rightarrow} s' \text{ in } \langle S, A, \alpha'\rangle = \Phi(\langle S, A, \alpha\rangle)$$

i.e.,

$$\langle a, s'\rangle \in \alpha(s) \quad \Longleftrightarrow \quad \langle a, s'\rangle \in \alpha'(s).$$

Theorem 5.2.3. *The assignment Φ from Definition 5.2.2 is a $*$-translation.*

Proof We need to prove that Φ is injective and reflects and preserves bisimilarity. Let $\Phi(\langle S, A, \alpha\rangle) = \langle S, A^*, \alpha'\rangle$, $\Phi(\langle S, A, \beta\rangle) = \langle S, A^*, \beta'\rangle$ and $\alpha' = \beta'$. Then

$$\langle a, s'\rangle \in \alpha(s) \iff \langle a, s'\rangle \in \alpha'(s) \iff \langle a, s'\rangle \in \beta'(s) \iff \langle a, s'\rangle \in \beta(s).$$

Hence for any state s, $\alpha(s) = \beta(s)$, i.e., $\alpha = \beta$.

For the reflection of bisimilarity, let $s \sim t$ in $\Phi(\langle S, A, \alpha\rangle) = \langle S, A^*, \alpha'\rangle$. Hence there exists an equivalence bisimulation relation R such that $\langle s, t\rangle \in R$ and (according to Example 3.6.8) for all $w \in A^*$,

if $s \stackrel{w}{\Rightarrow} s'$ then there exists $t' \in S$ such that $t \stackrel{w}{\Rightarrow} t'$ and $\langle s', t'\rangle \in R$.

Assume $s \xrightarrow{a} s'$ in $\langle S, A, \alpha\rangle$. Then $s \stackrel{a}{\Rightarrow} s'$ in $\langle S, A, \alpha'\rangle$ and therefore there exists $t' \in S$ with $\langle s', t'\rangle \in R$ and $t \stackrel{a}{\Rightarrow} t'$, i.e., $t \xrightarrow{a} t'$. Hence, R is a bisimulation on $\langle S, A, \alpha\rangle$ i.e. $s \sim t$ in the original system.

For the preservation of bisimulation, let $s \sim t$ in $\langle S, A, \alpha\rangle$ and let R be an equivalence bisimulation relation such that $\langle s, t\rangle \in R$. Assume $s \stackrel{w}{\Rightarrow} s'$, for some word $w \in A^*$. We show by induction on the length of w that there exists t' with $t \stackrel{w}{\Rightarrow} t'$ and $\langle s', t'\rangle \in R$. If w has length 0, then $w = \varepsilon$, $s' = s$ and we take $t' = t$. Assume w has length $k + 1$, i.e. $w = a \cdot w'$ for $a \in A, w' \in A^*$. Pick s'' such that $s \xrightarrow{a} s'' \stackrel{w'}{\Rightarrow} s'$. Since $\langle s, t\rangle \in R$ we can pick t'' such that $t \xrightarrow{a} t''$ and $\langle s'', t''\rangle \in R$. By the inductive hypothesis, for w' we can choose t' such that $t'' \stackrel{w'}{\Rightarrow} t'$ and $\langle s', t'\rangle \in R$. Note that $t \xrightarrow{a} t'' \stackrel{w'}{\Rightarrow} t'$, i.e., $t \stackrel{w}{\Rightarrow} t'$. Hence R is a bisimulation on $\langle S, A^*, \alpha'\rangle$ and $s \sim t$ holds in the $*$-extension. □

Note that if $\mathcal{T}$ is a functor induced by a natural transformation η and if $\langle S, A, \alpha\rangle, \langle S, A, \beta\rangle$ are two systems such that, for some $s \in S$, $\alpha(s) = \beta(s)$, then, clearly,

$$\alpha'(s) = \eta_S(\alpha(s)) = \eta_S(\beta(s)) = \beta'(s) \tag{5.2}$$

for $\langle S, A, \alpha'\rangle = \mathcal{T}(\langle S, A, \alpha\rangle$, $\langle S, A, \beta'\rangle = \mathcal{T}(\langle S, A, \beta\rangle$.

However, the following simple example shows that the $*$-translation Φ from Definition 5.2.2 violates (5.2), and therefore it can not be induced by a natural transformation.

Example 5.2.4. Let $S = \{s_1, s_2, s_3\}$ and $A = \{a, b, c\}$. Consider the LTSs:

$$\langle S, A, \alpha\rangle : s_1 \xrightarrow{a} s_2 \xrightarrow{b} s_3 \quad \text{and} \quad \langle S, A, \beta\rangle : s_1 \xrightarrow{a} s_2 \xrightarrow{c} s_3.$$

Obviously $\alpha(s_1) = \beta(s_1)$. However, $\alpha'(s_1) = \{\langle \varepsilon, s_1\rangle, \langle a, s_2\rangle, \langle ab, s_3\rangle\}$ while $\beta'(s_1) = \{\langle \varepsilon, s_1\rangle, \langle a, s_2\rangle, \langle ac, s_3\rangle\}$.

We next show that the coalgebraic and the concrete definitions coincide in the case of LTS.

Theorem 5.2.5. *Let $\langle S, A, \alpha\rangle$ be an LTS. Let $\tau \in A$ be an invisible action and $s, t \in S$ any two states. Then $s \approx_{\{\tau\}} t$ according to Definition 5.1.3 w.r.t the pair $\langle \Phi, \{\tau\}\rangle$ if and only if $s \approx_l t$ according to Definition 5.2.1.*

Proof Assume $s \approx_{\{\tau\}} t$ for $s, t \in S$ of an LTS $\langle S, A, \alpha\rangle$. This means that $s \sim t$ in the LTS $\langle S, A_{\{\tau\}}, \eta_S^{\{\tau\}} \circ \alpha'\rangle$, i.e., there exists an equivalence bisimulation R on

this system with $\langle s,t\rangle \in R$. As usual, α' is such that $\langle S, A^*, \alpha'\rangle = \Phi(\langle S, A, \alpha\rangle)$. Here we have $\eta_S^{\{\tau\}} = \mathcal{L}(h_{\{\tau\}}, id_S) = \mathcal{P}(h_{\{\tau\}}, id_S)$ and

$$\begin{aligned}(\eta_S^{\{\tau\}} \circ \alpha')(s) &= \eta_S^{\{\tau\}}(\alpha'(s)) \\ &= \mathcal{P}(\langle h_{\{\tau\}}, id_S\rangle)(\alpha'(s)) \\ &= \{\langle h_{\{\tau\}}(w), s'\rangle \mid \langle w, s'\rangle \in \alpha'(s)\} \\ &= \{\langle u, s'\rangle \mid \exists w \in B_u \colon s \stackrel{w}{\Rightarrow} s'\}\end{aligned}$$

We denote the transition relation of the weak-τ-system $\langle S, A_{\{\tau\}}, \eta_S^{\{\tau\}} \circ \alpha'\rangle$ by $\Rightarrow_\tau$. The above shows that for any word $w = a_1 \dots a_k \in A_\tau$

$$s \stackrel{w}{\Rightarrow}_\tau s' \iff \langle w, s'\rangle \in (\eta_S^{\{\tau\}} \circ \alpha')(s) \iff \exists v \in B_w = \tau^* a_1 \tau^* \dots \tau^* a_k \tau^* \colon s \stackrel{v}{\Rightarrow} s'.$$

We will show that the relation R is a weak bisimulation on $\langle S, A, \alpha\rangle$ according to Definition 5.2.1. Let $s \xrightarrow{a} s'$ $(a \neq \tau)$. Then $s \stackrel{a}{\Rightarrow} s'$, implying $s \stackrel{a}{\Rightarrow}_\tau s'$. Since R is a bisimulation on the weak-τ-system, there exists t' such that $t \stackrel{a}{\Rightarrow}_\tau t'$ and $\langle s', t'\rangle \in R$. We only need to note here that $\stackrel{a}{\Rightarrow}_\tau = \xrightarrow{\tau}{}^* \circ \xrightarrow{a} \circ \xrightarrow{\tau}{}^*$. In case $s \xrightarrow{\tau} s'$ we have $s \stackrel{\tau}{\Rightarrow} s'$ implying now $s \stackrel{\varepsilon}{\Rightarrow}_\tau s'$. Hence, there exists t' such that $t \stackrel{\varepsilon}{\Rightarrow}_\tau t'$ and $\langle s', t'\rangle \in R$. Since $\stackrel{\varepsilon}{\Rightarrow}_\tau = \xrightarrow{\tau}{}^*$, we have proved that R is a weak bisimulation on $\langle S, A, \alpha\rangle$ according to Definition 5.2.1.

For the opposite, let R be a weak bisimulation on $\langle S, A, \alpha\rangle$ according to Definition 5.2.1 such that $\langle s,t\rangle \in R$. It is easy to show by induction that for all $\langle s,t\rangle \in R$ and for any $a \in A$, if $s \xrightarrow{\tau}{}^* \circ \xrightarrow{a} \circ \xrightarrow{\tau}{}^* s'$ then there exists t' such that $t \xrightarrow{\tau}{}^* \circ \xrightarrow{a} \circ \xrightarrow{\tau}{}^* t'$ and $\langle s', t'\rangle \in R$. Hence, if $s \stackrel{a}{\Rightarrow}_\tau s'$ then there exists t' with $t \stackrel{a}{\Rightarrow}_\tau t'$ and $\langle s', t'\rangle \in R$. Based on this, another simple inductive argument on k leads to the conclusion that for any word $w = a_1 \dots a_k \in A_\tau$, if $s \stackrel{w}{\Rightarrow}_\tau s'$ then there exists a t' such that $t \stackrel{w}{\Rightarrow}_\tau t'$ and $\langle s', t'\rangle \in R$, i.e. R is a bisimulation on the weak-τ-system and hence $s \approx_{\{\tau\}} t$. □

5.3 Weak bisimulation for generative systems

In this section we deal with generative systems and their weak bisimilarity. We first focus on the concrete definition of weak bisimulation by Baier and Hermanns [BH97, Bai98, BH99]. Inspired by it, we provide a functor that suits for a definition of a $*$-translation for generative systems. This way we obtain a coalgebraic definition of weak bisimulation for this type of systems. We show that our definition, although at first sight much stronger, coincides with the definition of Baier and Hermanns. Unlike in the case of LTSs, here the $*$-translation leaves the class of generative systems.

We have dealt with generative systems in the previous chapters, and we have seen that they can also be cast into action-type coalgebras. A different notation than the one via a transition function is also possible and common in the literature. In this section we will interchangeably use both notations.

Remark 5.3.1. A generative probabilistic system can also be defined as a triple $\langle S, A, \mathrm{P}\rangle$ where S and A are sets and $\mathrm{P} : S \times A \times S \to [0,1]$ with the property that for $s \in S$,

$$\sum_{a\in A,\ s'\in S} \mathrm{P}(s,a,s') \in \{0,1\}. \tag{5.3}$$

We speak of P as the probabilistic transition relation. Condition (5.3) states that for all $s \in S$, $\mathrm{P}(s, \_, \_)$ is either a distribution over $A \times S$ or $\mathrm{P}(s, \_, \_) \equiv 0$, i.e. s is a terminating state. As usual one writes $s \xrightarrow{a[p]} s'$ whenever $\mathrm{P}(s,a,s') = p$, and $s \xrightarrow{a} s'$ for $\mathrm{P}(s,a,s') > 0$.

The definition of bisimulation for generative systems can also be reformulated differently. Let $\langle S, A, \mathrm{P}\rangle$ be a generative system. An equivalence relation $R \subseteq S \times S$ is a (strong) bisimulation on $\langle S, A, \mathrm{P}\rangle$ if and only if $\langle s,t\rangle \in R$ implies that for all $a \in A$ and for all equivalence classes $C \in S/R$

$$\mathrm{P}(s,a,C) = \mathrm{P}(t,a,C). \tag{5.4}$$

Here we have put $\mathrm{P}(s,a,C) = \sum_{s'\in C} \mathrm{P}(s,a,s')$. Two states s and t are bisimilar if and only if they are related by some bisimulation relation.

This section is divided into several parts that lead to the correspondence result: First we introduce paths in a generative system and establish some notions and properties of paths. Next we define a measure on the set of paths, where we basically follow the lines of Baier and Hermanns [BH99, Bai98]. Furthermore, we present the definition of weak bisimulation by Baier and Hermanns, and we prove some properties of weak bisimulation relations that will be used later on (without restricting to finite state systems). Then we define a translation and prove that it is a $*$-translation which therefore provides us with a notion of weak-τ-bisimulation. The final part is devoted to the proof of correspondence of the notion of weak-τ-bisimulation defined by means of the given $*$-translation and the concrete notion by Baier and Hermanns.

5.3.1 Paths and cones in a generative system

Let $\langle S, A, \mathrm{P}\rangle$ be a generative system. A finite path π of $\langle S, A, \mathrm{P}\rangle$ is an alternating sequence $\langle s_0, a_1, s_1, a_2, \ldots, a_k, s_k\rangle$, where $k \in \mathbb{N}_0$, $s_i \in S$, $a_i \in A$, and $\mathrm{P}(s_{i-1}, a_i, s_i) > 0$, $i = 1, \ldots, k$. We will denote a finite path $\pi = \langle s_0, a_1, s_1, a_2, \ldots, a_k, s_k\rangle$ more suggestively by

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots s_{k-1} \xrightarrow{a_k} s_k\,.$$

Moreover, set

$$\mathrm{length}(\pi) = k,\ \mathrm{first}(\pi) = s_0,\ \mathrm{last}(\pi) = s_k,\ \mathrm{trace}(\pi) = a_1 a_2 \cdots a_k\,.$$

The path $\varepsilon_{s_0} = (s_0)$ will be understood as the empty path starting at s_0. We will often write just ε for an arbitrary empty path. Similar to the finite case,

an infinite path π of $\langle S, A, \mathrm{P}\rangle$ is an infinite sequence $\langle s_0, a_1, s_1, a_2, \ldots\rangle$, where $s_i \in S$, $a_i \in A$ and $\mathrm{P}(s_{i-1}, a_i, s_i) > 0$, $i \in \mathbb{N}$, and will be written as

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots$$

Again we set $\mathrm{first}(\pi) = s_0$. A path π is called complete if it is either infinite or it is finite with $\mathrm{last}(\pi)$ a terminating state.

The sets of all (finite or infinite) paths, of all finite paths and of all complete paths will be denoted by Paths, FPaths and CPaths, respectively. Moreover, if $s \in S$, we write

$$\begin{aligned}\mathrm{Paths}(s) &= \{\pi \in \mathrm{Paths} \mid \mathrm{first}(\pi) = s\},\\ \mathrm{FPaths}(s) &= \{\pi \in \mathrm{FPaths} \mid \mathrm{first}(\pi) = s\},\\ \mathrm{CPaths}(s) &= \{\pi \in \mathrm{CPaths} \mid \mathrm{first}(\pi) = s\}.\end{aligned}$$

The set $\mathrm{Paths}(s)$ is partially ordered in a natural way by the prefix relation which is defined as follows. For $\pi, \pi' \in \mathrm{Paths}(s)$ we have $\pi \preceq \pi'$ if and only if one of (a), (b) or (c) holds:

(a) Both, π and π', are finite, say $\pi \equiv s \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_k} s_k$, $\pi' \equiv s \xrightarrow{a'_1} s'_1 \cdots \xrightarrow{a'_n} s'_n$, and we have

$$k \leq n \text{ and } s_i = s'_i, a_i = a'_i,\ i \leq k\,.$$

(b) π is a finite and π' an infinite path, say $\pi \equiv s \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_k} s_k$, $\pi' \equiv s \xrightarrow{a'_1} s'_1 \xrightarrow{a'_2} s'_2 \cdots$, and we have

$$s_i = s'_i, a_i = a'_i,\ i \leq k\,.$$

(c) $\pi = \pi'$.

The complete paths are exactly the maximal elements in this partial order. For every $\pi \in \mathrm{Paths}(s)$, there exists a $\pi' \in \mathrm{CPaths}(s)$ such that $\pi \preceq \pi'$.

The following statement will be used at several occasions throughout this section.

Lemma 5.3.2. *For any state $s \in S$, the set* $\mathrm{FPaths}(s)$ *is at most countable.*

Proof We first show, by induction on the length of paths, that for any fixed natural number k the number of finite paths that start in s and have length k is at most countable. For $k = 1$ the statement follows from the fact that $\mathrm{P}(s, \_, \_)$ is a probability distribution on $A \times S$ which implies that it has at most countable support set (Proposition 2.1.2), i.e. $\mathrm{P}(s, a, s') > 0$ for at most countably many pairs $\langle a, s'\rangle \in A \times S$. Consider paths of length $n + 1$. By the inductive hypothesis there are at most countably many paths of length n. Each of these can be extended to a path of length $n + 1$ in at most countably many

ways, hence the number of paths of length $n+1$ is also countable. Finally, the statement follows since $\mathrm{FPaths}(s) = \bigcup_{k\in\mathbb{N}_0}\{\pi \in \mathrm{FPaths}(s) \mid \mathrm{length}(\pi) = k\}$. □

For a finite path $\pi \in \mathrm{FPaths}(s)$, let $\pi\uparrow$ denote the set

$$\pi\uparrow = \{\xi \in \mathrm{CPaths}(s) \mid \pi \preceq \xi\}$$

also called the cone of complete paths generated by the finite path π.

Note that always $\pi\uparrow \neq \emptyset$. Let

$$\Gamma = \big\{\pi\uparrow \mid \pi \in \mathrm{FPaths}(s)\big\} \subseteq \mathcal{P}(\mathrm{CPaths}(s))$$

denote the set of all cones. By Lemma 5.3.2 this set is at most countable. For the study of weak bisimulation in generative systems a thorough understanding of the geometry of cones is crucial. First of all let us state the following elementary property:

Lemma 5.3.3. *Let $\pi_1, \pi_2 \in \mathrm{FPaths}(s)$. Then the cones $\pi_1\uparrow$ and $\pi_2\uparrow$ are either disjoint or one is a subset of the other. In fact,*

$$\pi_1\uparrow \cap \pi_2\uparrow = \begin{cases} \pi_2\uparrow & \text{if } \pi_1 \preceq \pi_2 \\ \pi_1\uparrow & \text{if } \pi_2 \preceq \pi_1 \\ \emptyset & \text{if } \pi_1 \not\preceq \pi_2 \text{ and } \pi_2 \not\preceq \pi_1 \end{cases}$$

Moreover, we have $\pi_1\uparrow = \pi_2\uparrow$ if and only if either

$$\pi_1 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k,\ \pi_2 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k \xrightarrow{a_{k+1}} s_{k+1} \cdots \xrightarrow{a_n} s_n \tag{5.5}$$

and thereby

$$\mathrm{P}(s_{i-1}, a_i, s_i) = 1,\ i = k+1, \ldots, n \tag{5.6}$$

or vice-versa.

Proof Let $\hat{\pi} \in \pi_1\uparrow \cap \pi_2\uparrow$, $\hat{\pi} \in \mathrm{CPaths}(s)$. Then $\pi_1 \preceq \hat{\pi}$ and $\pi_2 \preceq \hat{\pi}$. This, by the definition of the prefix ordering, implies that $\pi_1 \preceq \pi_2$ or $\pi_2 \preceq \pi_1$. Assume $\pi_1 \preceq \pi_2$. Then

$$\pi \in \pi_2\uparrow \iff \pi_2 \preceq \pi \implies \pi_1 \preceq \pi \iff \pi \in \pi_1\uparrow$$

i.e., $\pi_2\uparrow \subseteq \pi_1\uparrow$ and therefore $\pi_1\uparrow \cap \pi_2\uparrow = \pi_2\uparrow$.

It is clear that (5.5) and (5.6) imply $\pi_1\uparrow = \pi_2\uparrow$. Assume $\pi_1\uparrow = \pi_2\uparrow$. Then $\pi_1\uparrow \cap \pi_2\uparrow \neq \emptyset$ and therefore $\pi_1 \preceq \pi_2$ or $\pi_2 \preceq \pi_1$. Assume $\pi_1 \preceq \pi_2$, $\pi_1 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k$, $\pi_2 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k \xrightarrow{a_{k+1}} s_{k+1} \cdots \xrightarrow{a_n} s_n$. If for some $i \in \{k+1, \ldots, n\}$ it happens that $\mathrm{P}(s_{i-1}, a_i, s_i) < 1$, then there exists an action $a_i' \in A$ and a state $s_i' \in S$ such that $\langle a_i', s_i'\rangle \neq \langle a_i, s_i\rangle$ and

$$\pi_2' \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_{i-1}} s_{i-1} \xrightarrow{a_i'} s_i'$$

is a path in $\langle S, A, \mathrm{P}\rangle$. Since $i \geq k+1$ we have $\pi_1 \preceq \pi_2'$. However, this path is not prefix related to π_2, i.e., we have $\pi_2' \not\preceq \pi_2$ and $\pi_2 \not\preceq \pi_2'$. Therefore $\pi_2'\uparrow \ \cap \pi_1\uparrow = \pi_2'\uparrow$ and $\pi_2'\uparrow \ \cap \pi_2\uparrow = \emptyset$ contradicting $\pi_1\uparrow \ = \ \pi_2\uparrow$. □

Let $\Pi \subseteq \mathrm{FPaths}(s)$. We say that Π is minimal if for any two $\pi_1, \pi_2 \in \Pi$, $\pi_1 \neq \pi_2$, we have $\pi_1\uparrow \ \cap \pi_2\uparrow = \emptyset$. Hence in a minimal set of paths Π no path of Π is a proper prefix of another path of Π. We will express that Π is minimal by writing $\min(\Pi)$. As example note that every singleton set $\{\pi\}$, $\pi \in \mathrm{FPaths}(s)$, is minimal.

For $\Pi \subseteq \mathrm{FPaths}(s)$ we denote by $\Pi\uparrow$ the set

$$\Pi\uparrow = \bigcup_{\pi\in\Pi} \pi\uparrow \ .$$

Then the fact $\min(\Pi)$ just means that $\Pi\uparrow$ is actually the disjoint union of all $\pi\uparrow$, $\pi \in \Pi$, i.e.

$$\min(\Pi) \quad \Longleftrightarrow \quad \Pi\uparrow = \bigsqcup_{\pi\in\Pi} \pi\uparrow \ ,$$

where, here and in the sequel, the symbol $\sqcup$ denotes disjoint union. It is an immediate consequence of the definition that,

$$\min(\Pi), \ \Pi' \subseteq \Pi \quad \Longrightarrow \quad \min(\Pi').$$

If Π_1 and Π_2 are minimal, their union need not necessarily be minimal, even if $\Pi_1 \cap \Pi_2 = \emptyset$. We will use the notation

$$\Pi = \biguplus_{i\in I} \Pi_i$$

to express that

$$\Pi_i \subseteq \mathrm{FPaths}(s),\ i \in I, \ \Pi = \bigsqcup_{i\in I} \Pi_i \text{ and } \min(\Pi)\,.$$

Note that if $\Pi = \biguplus_{i\in I} \Pi_i$, also $\min(\Pi_i)$ for all $i \in I$. In particular this notation applies to minimal subsets Π written as the union of their one-element subsets:

$$\min(\Pi) \quad \Longrightarrow \quad \Pi = \biguplus_{\pi\in\Pi} \{\pi\}.$$

Observe that the following properties hold:

1. If $\Pi = \biguplus_{i\in I} \Pi_i$, then

$$\Pi\uparrow = \bigsqcup_{i\in I} \Pi_i\uparrow = \bigsqcup_{i\in I, \pi\in\Pi_i} \pi\uparrow \ .$$

2. We have $\Pi = \biguplus_{i\in I} \Pi_i$ if and only if
   (a) $\forall i \in I :\ \min(\Pi_i)$,
   (b) $\forall i,j \in I :\ i \neq j \implies \Pi_i \cap \Pi_j = \emptyset$, and
   (c) $\forall i,j \in I :\ i \neq j \implies \forall \pi_i \in \Pi_i, \forall \pi_j \in \Pi_j :\ \pi_i \not\preceq \pi_j, \pi_j \not\preceq \pi_i$.

Lemma 5.3.4. *Let* $\Pi \subseteq \mathrm{FPaths}(s)$. *Then there exists a unique set* $\Pi\!\downarrow \subseteq \mathrm{FPaths}(s)$, *such that*

(*i*) $\Pi\!\downarrow \subseteq \Pi$, $\min(\Pi\!\downarrow)$, *and*

$$\Pi\!\uparrow = \Big(\Pi\!\downarrow\Big)\!\uparrow\ .$$

(*ii*) *For every set* $\Pi' \subseteq \mathrm{FPaths}(s)$ *which possesses the property* (*i*), *we have*

$$\forall\, \pi' \in \Pi',\ \exists\, \pi \in \Pi\!\downarrow :\ \pi \preceq \pi'.$$

Proof Let $\Pi \subseteq \mathrm{FPaths}(s)$. Take

$$\Pi\!\downarrow = \{\pi \in \Pi \mid \forall \pi' \in \Pi : \pi' \not\prec \pi\}.$$

If $\Pi \neq \emptyset$, then $\Pi\!\downarrow \neq \emptyset$ since $\prec$ is an order relation and there are no infinite prefix descending sequences. The set $\Pi\!\downarrow$ is minimal by construction. Also $\Pi\!\downarrow \subseteq \Pi$ implying $(\Pi\!\downarrow)\!\uparrow \subseteq \Pi\!\uparrow$. Moreover $\forall \pi \in \Pi, \exists \pi' \in \Pi\!\downarrow : \pi' \preceq \pi$. Hence, by Lemma 5.3.3, for any $\pi \in \Pi$, there exists $\pi' \in \Pi\!\downarrow$ such that $\pi\!\uparrow \subseteq \pi'\!\uparrow$ i.e. $\Pi\!\uparrow \subseteq (\Pi\!\downarrow)\!\uparrow$ and we have shown (*i*).

Let Π' be a set that satisfies (*i*), i.e., $\Pi' \subseteq \Pi$, $\min(\Pi')$ and $\Pi\!\uparrow = \Pi'\!\uparrow$. Let $\pi' \in \Pi'$. Then $\pi' \in \Pi$ and as noted before there exists $\pi \in \Pi\!\downarrow$ such that $\pi \preceq \pi'$, proving (*ii*). The uniqueness follows from (*ii*) and the minimality of $\Pi\!\downarrow$. □

5.3.2 The measure *Prob*

Our first task is to construct out of P a probability measure on a certain σ-algebra on $\mathrm{CPaths}(s)$. This method was used in [BH99, Bai98], and before that in [Seg95b]. However, for the convenience of the reader we shall give complete proofs. As a standard reference for measure theoretic notions and results we use the monograph [Zaa58]. A famous measure theoretic theorem is the extension theorem which states that any pre-measure (σ-additive, monotone function with value zero for the empty set) on a semi-ring extends in a unique way to a measure on the σ-field generated by the semi-ring. Slightly different versions of this theorem apply to different definitions of the notion "semi-ring". For our purposes, the definition of a semi-ring from [Zaa58] fits best. Namely, a family of subsets of a given set S is a semi-ring if it contains the empty set, is closed under intersection and the set difference of any two of its elements is a disjoint union of at most countably many elements of the semi-ring. Therefore we refer to [Zaa58] rather than to more popular texts such as [Hal50].

Lemma 5.3.5. *The set* $\Gamma \cup \{\emptyset\}$ *is a semi-ring.*

Proof Clearly, $\Gamma \cup \{\emptyset\}$ contains the empty set and it is closed under intersection, by Lemma 5.3.3. We need to check that the set-difference of any two of its elements is a disjoint union of at most countably many elements of $\Gamma \cup \{\emptyset\}$. Let $\pi_1\uparrow, \pi_2\uparrow \in \Gamma$. We consider $\pi_1\uparrow \setminus \pi_2\uparrow$. Since $\pi_1\uparrow \setminus \pi_2\uparrow = \pi_1\uparrow \setminus (\pi_1\uparrow \cap \pi_2\uparrow)$, by Lemma 5.3.3, the only interesting case is $\pi_1\uparrow \cap \pi_2\uparrow = \pi_2\uparrow$ i.e. $\pi_1 \preceq \pi_2$.

Let

$$\pi_1 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k,\ \pi_2 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k \xrightarrow{a_{k+1}} s_{k+1} \cdots \xrightarrow{a_n} s_n$$

and put

$$\Pi = \{\pi \mid \pi \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_m} s_m \xrightarrow{a} s',\ k \leq m{<}n,\ s \xrightarrow{a_1} \cdots \xrightarrow{a_m} s_m \prec \pi_2, \pi \not\prec \pi_2\}.$$

Then $\pi_1\uparrow \setminus \pi_2\uparrow = \Pi\uparrow = \cup_{\pi\in\Pi} \pi\uparrow$ and the union is at most countable since the set Π is at most countable by Lemma 5.3.2. □

Now we are ready to introduce the desired extension of P to a measure. By Lemma 5.3.3 a function $\mathrm{Prob} : \Gamma \cup \{\emptyset\} \to [0,1]$ is well defined by

$$\mathrm{Prob}(C) = \begin{cases} \mathrm{P}(s,a_1,s_1)\cdot\ldots\cdot\mathrm{P}(s_{k-1},a_k,s_k) & , \begin{array}{l} C = \pi\uparrow \text{ with } k \geq 1, \\ \pi = s \xrightarrow{a_1} s_1 \cdots s_{k-1} \xrightarrow{a_k} s_k \end{array} \\ 1 & , C = \varepsilon\uparrow = \mathrm{CPaths}(s) \\ 0 & , C = \emptyset \end{cases}$$

Lemma 5.3.6. *The function* Prob *is a pre-measure*[1] *on the semi-ring* $\Gamma \cup \{\emptyset\}$.

Proof The proof of this property is a reformulation of the proof by Segala [Seg95b]. By definition $\mathrm{Prob}(\emptyset) = 0$. We need to check σ-additivity and monotonicity.

For the σ-additivity, assume

$$\pi\uparrow = \bigsqcup_{i\in I} \pi_i\uparrow \tag{5.7}$$

for some at most countable index set I. We need to show that $\mathrm{Prob}(\pi\uparrow) = \sum_{i\in I} \mathrm{Prob}(\pi_i\uparrow)$.

If π is a terminating path, then $|I| = 1$ and the property is trivially satisfied. Therefore we assume that π is not terminating and that $|I|{>}1$. Let $\{\pi_o \mid o \in O\}$ be the set of paths that extend π in one step, which means that

$$\forall o \in O\colon\ \pi \prec \pi_o, \mathrm{length}(\pi_o) = \mathrm{length}(\pi) + 1. \tag{5.8}$$

Then

$$\pi\uparrow = \bigsqcup_{o\in O} \pi_o\uparrow \tag{5.9}$$

[1] In [Zaa58] pre-measures are also called measures.

and

$$\begin{aligned}\sum_{o\in O}\mathrm{Prob}(\pi_o\uparrow) &= \sum_{a\in A,s'\in S}\mathrm{Prob}(\pi\uparrow)\cdot \mathrm{P}(\mathrm{last}(\pi),a,s')\\ &= \mathrm{Prob}(\pi\uparrow)\cdot\sum_{a\in A,s'\in S}\mathrm{P}(\mathrm{last}(\pi),a,s')\\ &= \mathrm{Prob}(\pi\uparrow) \qquad (5.10)\end{aligned}$$

since π does not end in a terminating state, i.e. $\sum_{a\in A,s\in S}\mathrm{P}(\mathrm{last}(\pi),a,s)=1$. By the assumption (5.7) we have that

$$\forall i\in I:\ \pi\preceq\pi_i. \qquad (5.11)$$

Moreover, from (5.7) and (5.9), using Lemma 5.3.3 we easily conclude that

$$\forall i\in I,\exists! o\in O:\ \pi_o\preceq\pi_i \qquad (5.12)$$

and

$$\forall o\in O,\exists i\in I:\ \pi_o\preceq\pi_i. \qquad (5.13)$$

Let

$$I_o=\{i\in I\mid \pi_o\preceq\pi_i\}.$$

From (5.12) and (5.13), we get that

$$I=\bigsqcup_{o\in O}I_o \qquad \text{and} \qquad \pi_o\uparrow=\bigsqcup_{i\in I_o}\pi_i\uparrow \quad \text{for } o\in O. \qquad (5.14)$$

We will now define a partial function, depth, that assigns to some finite paths an ordinal number, by a maximal assignment following the following rules:

If $\xi\in\mathrm{FPaths}(s)$ is such that $\pi_i\preceq\xi$ for some $i\in I$, then $\mathrm{depth}(\xi)=0$. If $\xi\in\mathrm{FPaths}(s)\cap\mathrm{CPaths}(s)$ i.e. ξ terminates, then also $\mathrm{depth}(\xi)=0$. If ξ is a finite path that has not yet assigned depth, but all its one step successors $\{\xi'\mid\xi\preceq\xi',\mathrm{length}(\xi')=\mathrm{length}(\xi)+1\}$ have assigned depth then put

$$\mathrm{depth}(\xi)=\sup\{\mathrm{depth}(\xi')\mid\xi\preceq\xi',\mathrm{length}(\xi')=\mathrm{length}(\xi)+1\}+1. \qquad (5.15)$$

We first show, by reducing to contradiction, that our starting finite path π has been assigned a value for depth. Assume that π has not been assigned a value for depth. Let $\pi^0=\pi$. For each $i>0$ let π^i be a path such that $\mathrm{length}(\pi^i)=\mathrm{length}(\pi^{i-1})+1$, $\pi^{i-1}\preceq\pi^i$ and π^i has not been assigned a value for depth. Such a chain under the prefix ordering exists since if for some i all paths that extend π^i in one step would had been assigned depth, then π^i would also have been assigned a depth. Consider the infinite complete path π^∞ such that for all $i>0$, $\pi^i\preceq\pi^\infty$. By definition $\pi^\infty\in\pi\uparrow$. By (5.7), there exists $i\in I$ such that $\pi^\infty\in\pi_i\uparrow$, implying that $\pi_i\preceq\pi^\infty$ and hence $\pi_i=\pi^n$ for some $n\geq 0$. However, then $\mathrm{depth}(\pi^n)=\mathrm{depth}(\pi_i)=0$ contradicting that π^n has no value for depth assigned.

Let $\mathrm{depth}(\pi) = \alpha$. If $\alpha = 0$ then $|I| = 1$ and σ-additivity trivially holds. Assume α is a successor or limit ordinal and assume that σ-additivity holds for any finite path ξ with $\mathrm{depth}(\xi) < \alpha$. Then we get

$$\begin{aligned}\mathrm{Prob}(\pi\uparrow) &\overset{(5.10)}{=} \sum_{o\in O} \mathrm{Prob}(\pi_o\uparrow) \\ &\overset{(I.H.)}{=} \sum_{o\in O}\sum_{i\in I_o} \mathrm{Prob}(\pi_i\uparrow) \\ &\overset{(5.14)}{=} \sum_{i\in I} \mathrm{Prob}(\pi_i\uparrow).\end{aligned}$$

where the inductive hypothesis is applicable since by (5.15) and (5.8), $\mathrm{depth}(\pi_o) < \alpha$ for all $o \in O$. This completes the proof of σ-additivity.

The function Prob is monotonic by definition: Assume $\pi_1\uparrow \subseteq \pi_2\uparrow$. Then, by Lemma 5.3.3, two things are possible. Either $\pi_2 \prec \pi_1$ and since $\mathrm{P}(s,a,t) \leq 1$ for all $s,t \in S, a \in A$, from the definition of Prob we get $\mathrm{Prob}(\pi_1\uparrow) \leq \mathrm{Prob}(\pi_2\uparrow)$, or $\pi_1\uparrow = \pi_2\uparrow$, in which case $\mathrm{Prob}(\pi_1\uparrow) = \mathrm{Prob}(\pi_2\uparrow)$. □

Corollary 5.3.7. *The function* Prob *extends uniquely to a probability measure on the σ-algebra on* $\mathrm{CPaths}(s)$ *generated by* $\Gamma \cup \{\emptyset\}$. *We will denote this measure again by* Prob.

Remark 5.3.8. Note that, although paths are more or less just sequences of elements of S and A, not only the function Prob itself, but also the σ-algebra where it is defined and in fact already the base set $\mathrm{CPaths}(s)$ depends heavily on P. At first sight this might seem to be an undesirable fact, however, a second look at the matters shows that it cannot be avoided.

The measure Prob induces a set-function on finite paths, which we will also denote by Prob. Define $\mathrm{Prob} : \mathcal{P}(\mathrm{FPaths}(s)) \to [0,1]$ by

$$\mathrm{Prob}(\Pi) = \mathrm{Prob}(\Pi\uparrow).$$

This notation is not in conflict with the already existing notation of the measure Prob. In fact, $\mathcal{P}(\mathrm{FPaths}(s)) \cap \mathcal{P}(\mathrm{CPaths}(s))$ consists entirely of Prob-measurable sets and on such sets both definitions coincide. To see this, note that if $\pi \in \mathrm{FPaths}(s) \cap \mathrm{CPaths}(s)$, then $\pi\uparrow = \{\pi\}$. Thus, if $\Pi \in \mathcal{P}(\mathrm{FPaths}(s)) \cap \mathcal{P}(\mathrm{CPaths}(s))$, we have

$$\Pi = \bigsqcup_{\pi\in\Pi} \{\pi\} = \bigsqcup_{\pi\in\Pi} \pi\uparrow = \Pi\uparrow,$$

and this union is at most countable.

It will always be clear from the context whether we mean the measure Prob or the just defined set-function Prob. Still, there is a word of caution in order:

The function $\mathrm{Prob} : \mathcal{P}(\mathrm{FPaths}(s)) \to [0,1]$ is in general not additive. However, looking at the notations introduced above, we find that

$$\Pi = \biguplus_{i\in I} \Pi_i \quad \Longrightarrow \quad \mathrm{Prob}(\Pi) = \sum_{i\in I} \mathrm{Prob}(\Pi_i)\,.$$

In particular, we obtain that $\mathrm{Prob}(\Pi) = \sum_{\pi\in\Pi} \mathrm{Prob}(\pi\uparrow)$ for every minimal set Π. Moreover, by Lemma 5.3.4, we always have

$$\mathrm{Prob}(\Pi) = \mathrm{Prob}(\Pi\downarrow)\,.$$

We next introduce some particular sets of paths. For $s \in S$, $S', S'' \subseteq S$ with $S' \subseteq S''$, and $W, W' \subseteq A^*$ with $W \subseteq W'$, denote

$$s \xrightarrow[\neg S'']{W}_{\neg W'} S' = \left\{\pi \in \mathrm{FPaths}(s) \mid \begin{array}{c} \mathrm{last}(\pi) \in S', \mathrm{trace}(\pi) \in W \\ \forall\, \xi \prec \pi : \mathrm{trace}(\xi) \in W' \Rightarrow \mathrm{last}(\xi) \notin S'' \end{array}\right\}$$

and write $\mathrm{Prob}(s, W, \neg W, S', \neg S'') = \mathrm{Prob}(s \xrightarrow[\neg S'']{W}_{\neg W'} S')$. Since $S' \subseteq S''$ and $W \subseteq W'$ we always have $\min(s \xrightarrow[\neg S'']{W}_{\neg W'} S')$. For notational convenience we will drop redundant arguments whenever possible. Put

$$\begin{aligned} s \xrightarrow{W}_{\neg W'} S' &= s \xrightarrow[\neg S']{W}_{\neg W'} S', \\ s \xrightarrow{W}_{\neg S''} S' &= s \xrightarrow[\neg S'']{W}_{\neg W} S', \\ s \xrightarrow{W} S' \qquad &= s \xrightarrow[\neg S']{W}_{\neg W} S'\,, \end{aligned} \tag{5.16}$$

and, correspondingly,

$$\begin{aligned} \mathrm{Prob}(s, W, \neg W', S') &= \mathrm{Prob}(s, W, \neg W', S', \neg S'), \\ \mathrm{Prob}(s, W, S', \neg S'') &= \mathrm{Prob}(s, W, \neg W, S', \neg S''), \\ \mathrm{Prob}(s, W, S') \qquad &= \mathrm{Prob}(s, W, \neg W, S', \neg S')\,. \end{aligned} \tag{5.17}$$

Note that

$$s \xrightarrow{W} S' = \big\{\pi \in \mathrm{FPaths}(s) \mid \mathrm{last}(\pi) \in S', \mathrm{trace}(\pi) \in W\big\}\downarrow\ .$$

Let S', S'', W, W' be as above and let moreover $F \subseteq S$ be given. Then denote

$$F \xrightarrow[\neg S'']{W}_{\neg W'} S' = \bigsqcup_{s\in F} s \xrightarrow[\neg S'']{W}_{\neg W'} S' \subseteq \mathrm{FPaths}$$

We will often encounter the situation that for every $s \in F$ the value of $\mathrm{Prob}(s, W, \neg W', S', \neg S'')$ is the same. In this case we speak of this value as $\mathrm{Prob}(F, W, \neg W', S', \neg S'')$. Also, in this context, we shall freely apply shortening of notation as in (5.16) and (5.17).

Next we define sets of concatenated paths. For $\Pi \subseteq \mathrm{FPaths}$, put

$$\mathrm{first}(\Pi) = \{\mathrm{first}(\pi) \mid \pi \in \Pi\},\ \mathrm{last}(\Pi) = \{\mathrm{last}(\pi) \mid \pi \in \Pi\}\,.$$

If $\Pi_1, \Pi_2 \subseteq \mathrm{FPaths}$ and $\mathrm{last}(\Pi_1) = \mathrm{first}(\Pi_2)$, we define

$$\Pi_1 \cdot \Pi_2 = \left\{ \pi_1 \cdot \pi_2 \mid \pi_1 \in \Pi_1, \pi_2 \in \Pi_2, \mathrm{last}(\pi_1) = \mathrm{first}(\pi_2) \right\},$$

where $\pi_1 \cdot \pi_2 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k \xrightarrow{a_{k+1}} \cdots \xrightarrow{a_n} s_n$ for $\pi_1 \equiv s \xrightarrow{a_1} \cdots \xrightarrow{a_k} s_k$ and $\pi_2 \equiv s_k \xrightarrow{a_{k+1}} \cdots \xrightarrow{a_n} s_n$. Note that, whenever a concatenation $\pi_1 \cdot \pi_2$ is defined, we have $\mathrm{Prob}(\{\pi_1 \cdot \pi_2\}) = \mathrm{Prob}(\{\pi_1\}) \cdot \mathrm{Prob}(\{\pi_2\})$.

The next technical proposition will be used at several occasions in this chapter.

Proposition 5.3.9. *Let* $\Pi_1 \subseteq \mathrm{FPaths}(s)$, $\Pi_2 \subseteq \mathrm{FPaths}$ *with* $\mathrm{last}(\Pi_1) = \mathrm{first}(\Pi_2)$ *and assume that this set is represented as a disjoint union*

$$\mathrm{last}(\Pi_1) = \mathrm{first}(\Pi_2) = \bigsqcup_{i \in I} S_i \,.$$

Denote $\Pi_{1,S_i} = \{\pi_1 \in \Pi_1 \mid \mathrm{last}(\pi_1) \in S_i\}$, $\Pi_{2,t} = \{\pi_2 \in \Pi_2 \mid \mathrm{first}(\pi_2) = t\}$. *Assume that for every* $i \in I$

$$\mathrm{Prob}(\Pi_{2,t'}) = \mathrm{Prob}(\Pi_{2,t''}),\ t', t'' \in S_i \,.$$

Moreover, assume that Π_1, Π_2 *and* $\Pi_1 \cdot \Pi_2$ *are minimal. Then, for every choice of* $(t_i)_{i \in I} \in \prod_{i \in I} S_i$, *we have*

$$\mathrm{Prob}(\Pi_1 \cdot \Pi_2) = \sum_{i \in I} \mathrm{Prob}(\Pi_{1,S_i}) \cdot \mathrm{Prob}(\Pi_{2,t_i}) \,.$$

Proof Denote by $\Pi_{2,S_i} = \{\pi_2 \in \Pi_2 \mid \mathrm{first}(\pi_2) \in S_i\}$ and by $\Pi_{1,t} = \{\pi_1 \in \Pi_1 \mid \mathrm{last}(\pi_1) = t\}$. Under the assumptions of the proposition, we have

$$\begin{aligned}
\mathrm{Prob}(\Pi_1 \cdot \Pi_2) &= \mathrm{Prob}\Big(\biguplus_{\pi \in \Pi_1 \cdot \Pi_2} \pi\uparrow \Big) \\
&= \mathrm{Prob}\Big(\biguplus_{i \in I} \Big(\biguplus_{\pi \in \Pi_{1,S_i} \cdot \Pi_{2,S_i}} \pi\uparrow \Big) \Big) \\
&= \mathrm{Prob}\Big(\biguplus_{i \in I} \Big(\biguplus_{t \in S_i} \Big(\biguplus_{\pi \in \Pi_{1,t} \cdot \Pi_{2,t}} \pi\uparrow \Big) \Big) \Big) \\
&= \sum_{i \in I} \sum_{t \in S_i} \sum_{\pi \in \Pi_{1,t} \cdot \Pi_{2,t}} \mathrm{Prob}(\pi\uparrow)
\end{aligned}$$

Since, by minimality, $\Pi_{1,t} \times \Pi_{2,t} \cong \Pi_{1,t} \cdot \Pi_{2,t}$ via $(\pi_1, \pi_2) \mapsto \pi_1 \cdot \pi_2$, we have

$$\begin{aligned}
\sum_{\pi \in \Pi_{1,t} \cdot \Pi_{2,t}} \mathrm{Prob}(\pi\uparrow) &= \sum_{(\pi_1, \pi_2) \in \Pi_{1,t} \times \Pi_{2,t}} \mathrm{Prob}(\pi_1 \cdot \pi_2\uparrow) \\
&= \sum_{\pi_1 \in \Pi_{1,t}} \sum_{\pi_2 \in \Pi_{2,t}} \mathrm{Prob}(\pi_1\uparrow) \mathrm{Prob}(\pi_2\uparrow) \\
&= \sum_{\pi_1 \in \Pi_{1,t}} \mathrm{Prob}(\pi_1\uparrow) \cdot \sum_{\pi_2 \in \Pi_{2,t}} \mathrm{Prob}(\pi_2\uparrow) \\
&= \mathrm{Prob}(\Pi_{1,t}) \cdot \mathrm{Prob}(\Pi_{2,t}) \,.
\end{aligned}$$

Since, by assumption, for every $i \in I$ the value of $\mathrm{Prob}(\Pi_{2,t})$ does not depend on $t \in S_i$, it follows that

$$\begin{aligned}\mathrm{Prob}(\Pi_1 \cdot \Pi_2) &= \sum_{i\in I} \sum_{t\in S_i} \mathrm{Prob}(\Pi_{1,t}) \cdot \mathrm{Prob}(\Pi_{2,t}) \\ &= \sum_{i\in I} \Big(\mathrm{Prob}(\Pi_{2,t_i}) \cdot \sum_{t\in S_i} \mathrm{Prob}(\Pi_{1,t}) \Big) \\ &= \sum_{i\in I} \mathrm{Prob}(\Pi_{2,t_i})\, \mathrm{Prob}(\Pi_{1,S_i})\,.\end{aligned}$$

□

It is worth to explicitly note the particular case of this proposition when $|I| = 1$.

Corollary 5.3.10. *Let* $\Pi_1 \subseteq \mathrm{FPaths}(s)$, $\Pi_2 \subseteq \mathrm{FPaths}$ *with* $\mathrm{last}(\Pi_1) = \mathrm{first}(\Pi_2)$. *Let* $\Pi_{2,t} = \{\pi_2 \in \Pi_2 \mid \mathrm{first}(\pi_2) = t\}$. *Then, if* $\min(\Pi_1)$, $\min(\Pi_2)$ *and* $\min(\Pi_1 \cdot \Pi_2)$, *and if for any* $t', t'' \in \mathrm{first}(\Pi_2)$, $\mathrm{Prob}(\Pi_{2,t'}) = \mathrm{Prob}(\Pi_{2,t''})$, *we have that*

$$\mathrm{Prob}(\Pi_1 \cdot \Pi_2) = \mathrm{Prob}(\Pi_1) \cdot \mathrm{Prob}(\Pi_{2,t})$$

for arbitrary $t \in \mathrm{first}(\Pi_2)$. □

For further reference, we state the following simple property.

Proposition 5.3.11. *Consider a generative system* $\langle S, A, \mathrm{P}\rangle$. *Let* $s \in S$, $W \subseteq A^*$ *and* $S' \subseteq S$ *such that it partitions as* $S' = \sqcup_{i\in I} S_i$. *Then*

$$\mathrm{Prob}(s, W, S') = \sum_{i\in I} \mathrm{Prob}(s, W, S_i, \neg S').$$

Proof It holds that

$$s \xrightarrow{W} S' = \biguplus_{i\in I} s \xrightarrow{W}_{\neg S'} S_i.$$

□

5.3.3 The concrete weak bisimulation

In this subsection we recall the original definition of weak bisimulation for generative systems by Baier and Hermanns and we establish some properties of the weak bisimulation relations that are essential for the correspondence result of Section 5.3.5 below.

Definition 5.3.12. *[BH97, Bai98, BH99] Let* $\langle S, A, \mathrm{P}\rangle$ *be a generative system. Let* $\tau \in A$ *be an invisible action. An equivalence relation* $R \subseteq S \times S$ *is a weak bisimulation on* $\langle S, A, \mathrm{P}\rangle$ *if and only if* $\langle s, t\rangle \in R$ *implies that for all actions* $a \in A \setminus \{\tau\}$ *and for all equivalence classes* $C \in S/R$:

$$\mathrm{Prob}(s, \tau^* a \tau^*, C) = \mathrm{Prob}(t, \tau^* a \tau^*, C) \tag{5.18}$$

and for all $C \in S/R$:

$$\mathrm{Prob}(s, \tau^*, C) = \mathrm{Prob}(t, \tau^*, C). \tag{5.19}$$

Two states s and t are weakly bisimilar if and only if they are related by some weak bisimulation relation. Notation $s \approx_g t$.

Note the analogy between the transfer conditions (5.18), (5.19) and (5.4). We borrow the next proposition from Baier and Hermanns, [Bai98, BH99].

Proposition 5.3.13. *Let $\langle S, A, \mathrm{P}\rangle$ be a generative system and let $s \approx_g t$. If R is a weak bisimulation relating s and t, then for all $a_1, \ldots, a_k \in A \setminus \{\tau\}$ and for all classes $C \in S/R$*

$$\mathrm{Prob}(s, \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^*, C) = \mathrm{Prob}(t, \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^*, C).$$

Proof Let R be a weak bisimulation on $\langle S, A, \mathrm{P}\rangle$ such that $\langle s, t\rangle \in R$. We prove the property by induction on k. For $k \in \{0, 1\}$ the property holds by Definition 5.3.12. Let $B = \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^*$. Assume $\mathrm{Prob}(s, B, C) = \mathrm{Prob}(t, B, C)$ for all $C \in S/R$ and let $B' = \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^* a_{k+1} \tau^*$. We have

$$s \xrightarrow{B'} C = \biguplus_{C' \in S/R} s \xrightarrow{B} C' \cdot C'^{\,\tau^* a_{k+1} \tau^*} \!\!\longrightarrow C$$

and, since R is a weak bisimulation, for any class $C' \in S/R$ and for any $t', t'' \in C'$ we have $\mathrm{Prob}(t', \tau^* a_{k+1} \tau^*, C) = \mathrm{Prob}(t'', \tau^* a_{k+1} \tau^*, C)$ and we may write this common value as $\mathrm{Prob}(C', \tau^* a_{k+1} \tau^*, C)$. Hence, we may apply Corollary 5.3.10 and we get,

$$\begin{aligned}
\mathrm{Prob}(s, B', C) &= \sum_{C' \in S/R} \mathrm{Prob}(s, B, C') \cdot \mathrm{Prob}(C', \tau^* a_{k+1} \tau^*, C) \\
&\overset{(IH)}{=} \sum_{C' \in S/R} \mathrm{Prob}(t, B, C') \cdot \mathrm{Prob}(C', \tau^* a_{k+1} \tau^*, C) \\
&= \mathrm{Prob}(t, B', C).
\end{aligned}$$

□

Let R be a weak bisimulation on $\langle S, A, \mathrm{P}\rangle$. Define a relation $\to$ on S/R by

$$C_1 \to C_2 \iff \mathrm{Prob}(C_1, \tau^*, C_2) = 1$$

and denote by $\leftrightarrow$ the equivalence closure of $\to$, i.e., $\leftrightarrow = (\to \cup \leftarrow)^*$.

A weak bisimulation on $\langle S, A, \mathrm{P}\rangle$ is called complete, if $\mathrm{Prob}(C_1, \tau^*, C_2) = 1 \iff C_1 = C_2$ for all classes $C_1, C_2 \in S/R$. Hence, if R is a complete weak bisimulation then for any two different classes $C_1, C_2 \in S/R$ it holds that $\mathrm{Prob}(C_1, \tau^*, C_2) < 1$.

The next result is also stated in [BH99] and is used for the correspondence result below.

Proposition 5.3.14. *Let $\langle S, A, \mathrm{P}\rangle$ be a generative system and let $s \approx_g t$. Then there exists a complete weak bisimulation R relating s and t.*

We will gradually build up the proof of Proposition 5.3.14, by a sequence of lemmas showing properties of the $\rightarrow$ relation.

Lemma 5.3.15. *The relation $\rightarrow$ corresponding to a weak bisimulation R is reflexive and transitive.*

Proof Reflexivity follows since $\varepsilon \in C \xrightarrow{\tau^*} C$ for any class C. Namely, this implies that $1 = \mathrm{Prob}(\varepsilon) \leq \mathrm{Prob}(C, \tau^*, C) \leq 1$ and hence $C \rightarrow C$ for any class C.

Assume $C_1 \rightarrow C_2$ and $C_2 \rightarrow C_3$. It is important to note that for any $s \in C_1$,

$$(s \xrightarrow{\tau^*} C_2 \cdot C_2 \xrightarrow{\tau^*} C_3)\uparrow \subseteq \bigcup\{\pi\uparrow \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) \in \tau^*, \mathrm{last}(\pi) \in C_3\}$$

since every cone that contributes to the left-hand-side also contributes to the right-hand-side. Now, as in the proof of Proposition 5.3.13, using Corollary 5.3.10 and the fact that $s \in C_1$, we get

$$\begin{aligned}\mathrm{Prob}((s \xrightarrow{\tau^*} C_2 \cdot C_2 \xrightarrow{\tau^*} C_3)\uparrow) &= \mathrm{Prob}(s \xrightarrow{\tau^*} C_2 \cdot C_2 \xrightarrow{\tau^*} C_3)\\ &= \mathrm{Prob}(s \xrightarrow{\tau^*} C_2) \cdot \mathrm{Prob}(C_2 \xrightarrow{\tau^*} C_3)\\ &= 1\end{aligned}$$

and

$$\mathrm{Prob}(\bigcup\{\pi\uparrow \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) \in \tau^*, \mathrm{last}(\pi) \in C_3\}) = \mathrm{Prob}(C_1 \xrightarrow{\tau^*} C_3).$$

Hence, $1 \leq \mathrm{Prob}(C_1 \xrightarrow{\tau^*} C_3)$, i.e. $\mathrm{Prob}(C_1 \xrightarrow{\tau^*} C_3) = 1$. □

We next investigate in more detail the behavior of the $\rightarrow$ relation.

Lemma 5.3.16. *Let R be a weak bisimulation on $\langle S, A, \mathrm{P}\rangle$. Let C_1, C_2, C_3 be different elements of S/R and assume $C_1 \rightarrow C_2$. Then either (i) or (ii) holds.*

- (i) $\forall \pi \in C_1 \xrightarrow{\tau^*} C_3, \exists \pi' \in C_1 \xrightarrow{\tau^*} C_2 : \pi' \prec \pi$,
  i.e. all τ^* paths from C_1 to C_3 pass C_2.
- (ii) $C_3 \rightarrow C_2$

Proof Assume $C_1 \rightarrow C_2$ and not (i). Let $\pi \in C_1 \xrightarrow{\tau^*} C_3$ be a path that does not pass C_2. Let $s = \mathrm{first}(\pi)$. Since $\mathrm{Prob}(s, \tau^*, C_2) = 1$, also

$$\mathrm{Prob}(\pi\uparrow \cup \biguplus_{\bar{\pi} \in s \xrightarrow{\tau^*} C_2} \bar{\pi}\uparrow) = 1$$

implying that, by additivity and $\mathrm{Prob}(\pi\uparrow) > 0$,

$$\pi\uparrow \cap \biguplus_{\bar{\pi} \in s \xrightarrow{\tau^*} C_2} \bar{\pi}\uparrow \neq \emptyset$$

i.e., there exists $\bar{\pi} \in s \xrightarrow{\tau^*} C_2$ such that $\pi\uparrow \cap \bar{\pi}\uparrow \neq \emptyset$ which implies that $\pi \prec \bar{\pi}$ or $\bar{\pi} \prec \pi$. The latter is excluded by assumption. Now,

$$\pi\uparrow \cup \biguplus_{\bar{\pi} \in s \xrightarrow{\tau^*} C_2} \bar{\pi}\uparrow = \left(\pi\uparrow \cup \biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow \right) \sqcup \biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow = \emptyset}} \bar{\pi}\uparrow \ .$$

Hence,

$$\mathrm{Prob}(\pi\uparrow \cup \biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow) + \mathrm{Prob}(\biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow = \emptyset}} \bar{\pi}\uparrow) = 1$$

and, on the other hand, since $\mathrm{Prob}(s, \tau^*, C_2) = 1$,

$$\mathrm{Prob}(\biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow) + \mathrm{Prob}(\biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow = \emptyset}} \bar{\pi}\uparrow) = 1$$

implying

$$\mathrm{Prob}(\pi\uparrow \cup \biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow) = \mathrm{Prob}(\biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow)$$

and, since for any $\bar{\pi} \in s \xrightarrow{\tau^*} C_2$ with $\bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset$ we have (as before) $\pi \prec \bar{\pi}$ i.e. $\bar{\pi}\uparrow \subseteq \pi\uparrow$, we get that

$$\mathrm{Prob}(\pi\uparrow) = \mathrm{Prob}(\pi\uparrow \cup \biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow) = \mathrm{Prob}(\biguplus_{\substack{\bar{\pi} \in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow \cap \pi\uparrow \neq \emptyset}} \bar{\pi}\uparrow). \tag{5.20}$$

Consider the set of paths that extend π to a path in $s \xrightarrow{\tau^*} C_2$

$$\Pi = \{\hat{\pi} = \bar{\pi} - \pi \mid \pi \cdot \hat{\pi} = \bar{\pi}, \bar{\pi} \in s \xrightarrow{\tau^*} C_2\}.$$

Recall that $\mathrm{last}(\pi) \in C_3$. Then

$$\Pi \subseteq \mathrm{last}(\pi) \xrightarrow{\tau^*} C_2$$

and therefore the set Π is minimal and

$$\Pi \subseteq C_3 \xrightarrow{\tau^*} C_2. \tag{5.21}$$

We have, for any $\hat{\pi} = \bar{\pi} - \pi \in \Pi$,

$$\mathrm{Prob}(\hat{\pi}) = \frac{\mathrm{Prob}(\bar{\pi})}{\mathrm{Prob}(\pi)}$$

and therefore

$$\begin{aligned}
\mathrm{Prob}(\Pi) &= \sum_{\hat{\pi}\in\Pi} \mathrm{Prob}(\hat{\pi}) \\
&= \frac{\sum_{\substack{\bar{\pi}\in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow\cap\pi\uparrow\neq\emptyset}} \mathrm{Prob}(\bar{\pi})}{\mathrm{Prob}(\pi)} \\
&\overset{(*)}{=} \frac{\mathrm{Prob}(\biguplus_{\substack{\bar{\pi}\in s \xrightarrow{\tau^*} C_2 \\ \bar{\pi}\uparrow\cap\pi\uparrow\neq\emptyset}} \bar{\pi}\uparrow\,)}{\mathrm{Prob}(\pi\uparrow\,)} \\
&\overset{(5.20)}{=} 1
\end{aligned}$$

where $(*)$ holds by the minimality of the set $\{\bar{\pi} \mid \bar{\pi} \in s \xrightarrow{\tau^*} C_2, \bar{\pi}\uparrow\ \cap \pi\uparrow \neq \emptyset\}$. Hence, by (5.21),

$$\mathrm{Prob}(C_3 \xrightarrow{\tau^*} C_2) \geq \mathrm{Prob}(\Pi) = 1,$$

i.e. $C_3 \to C_2$. □

The next lemma states that if a path exits a class C_1 with a trace that does not consist entirely of τ's, given that $C_1 \to C_2$, then this path must pass C_2 after performing a τ-trace.

Lemma 5.3.17. *Let R be a weak bisimulation on $\langle S, A, \mathrm{P}\rangle$. Let C_1, C_2 be different elements of S/R and assume $C_1 \to C_2$. If for $s \in C_1$, $\pi \in s \xrightarrow{\tau^* a \tau^*} S$, then there exists $\pi' \in C_1 \xrightarrow{\tau^*} C_2$ such that $\pi' \prec \pi$.*

Proof A similar argument as for Lemma 5.3.16 applies here as well. Assume $\pi \in s \xrightarrow{\tau^* a \tau^*} S$. Since $\mathrm{Prob}(s, \tau^*, C_2) = 1$, also

$$\mathrm{Prob}(\pi\uparrow\ \cup \biguplus_{\bar{\pi}\in s \xrightarrow{\tau^*} C_2} \bar{\pi}\uparrow\,) = 1$$

implying that

$$\pi\uparrow\ \cap \biguplus_{\bar{\pi}\in s \xrightarrow{\tau^*} C_2} \bar{\pi}\uparrow \neq \emptyset$$

i.e., there exists $\bar{\pi} \in s \xrightarrow{\tau^*} C_2$ such that $\pi\uparrow\ \cap \bar{\pi}\uparrow \neq \emptyset$ which implies that $\bar{\pi} \prec \pi$ (since $\pi \prec \bar{\pi}$ is excluded by the form of the traces). □

Our next lemma shows a sort of confluence of the $\to$ relation.

Lemma 5.3.18. *Let R be a weak bisimulation on $\langle S, A, P\rangle$. If $C_1 \to C_2$ and $C_1 \to C_3$, then $C_2 \to C_3$ or $C_3 \to C_2$, for all $C_1, C_2, C_3 \in S/R$.*

Proof From Lemma 5.3.16 we get that either $C_3 \to C_2$, or each path from C_1 to C_3 with a trace in τ^* passes C_2. Hence, in the later case, we have

$$C_1 \xrightarrow{\tau^*} C_3 \subseteq C_1 \xrightarrow{\tau^*} C_2 \cdot C_2 \xrightarrow{\tau^*} C_3$$

i.e.

$$\mathrm{Prob}(C_1, \tau^*, C_3) \leq \mathrm{Prob}(C_1, \tau^*, C_2) \cdot \mathrm{Prob}(C_2, \tau^*, C_3)$$

which leads to $1 \leq \mathrm{Prob}(C_2, \tau^*, C_3)$ i.e. $C_2 \to C_3$. □

Next we establish a "sink" property for two $\to$ connected classes.

Lemma 5.3.19. *Let R be a weak bisimulation on $\langle S, A, P\rangle$. If $C_1 \leftrightarrow C_2$, then there exists C such that $C_1 \to C$ and $C_2 \to C$.*

Proof We prove this by induction on the length of the sequence of $\to$ and $\leftarrow$ connecting C_1 and C_2. For a sequence of length 0, we have $C_1 = C_2$ and the statement holds trivially, by reflexivity, with $C = C_1 = C_2$. Assume $C_1 \leftrightarrow C_2$ via a sequence of $\to$ and $\leftarrow$ of length $k+1$. Then there is a C_3 such that $C_1 \leftrightarrow C_3$ via a sequence of $\to$ and $\leftarrow$ of length k, and, $C_2 \to C_3$ or $C_3 \to C_2$. By the inductive hypothesis, there exists C such that $C_1 \to C$ and $C_3 \to C$. Now, if $C_2 \to C_3$, then also, by transitivity, $C_2 \to C$. If, on the other hand, $C_3 \to C_2$, then since also $C_3 \to C$, by Lemma 5.3.18, we get either $C \to C_2$ implying $C_1 \to C_2$, or $C_2 \to C$. □

From Lemma 5.3.19, by induction on the number of elements, we obtain a sink for any finite set of $\to$ connected classes.

Lemma 5.3.20. *Let R be a weak bisimulation on $\langle S, A, P\rangle$. Let $F \subseteq S/R$ be a finite set of classes, with the property that for all $C_1, C_2 \in F$, $C_1 \leftrightarrow C_2$. Then there exists a class $C \in S/R$ such that for all $C' \in F$, $C' \to C$.* □

The next result shows that we can join $\to$ connected classes of a weak bisimulation and still obtain a weak bisimulation. In the sequel by $[C]_{\leftrightarrow}$ we denote the $\leftrightarrow$ - equivalence class of C.

Lemma 5.3.21. *Let R be a weak bisimulation on $\langle S, A, P\rangle$. Let $C_0 \in S/R$ be a fixed class such that $U = [C_0]_{\leftrightarrow} \neq \{C_0\}$. Define an equivalence R' on S by determining the set of classes, as*

$$S/R' = \{C \in S/R \mid C \nleftrightarrow C_0\} \cup \{\cup_{C \in U} C\}.$$

Then R' is a weak bisimulation and $R \subset R'$.

Proof We need to prove that for all $a \in A$, all $K_1, K_2 \in S/R'$ and for all $s, t \in K_1$

$$\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, K_2) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, K_2)$$

where $\hat{a} = a$ if $a \neq \tau$ and $\hat{\tau} = \varepsilon$, the empty word. There are several cases:

Case 1. $K_1, K_2 \in S/R$.

The statement holds since R is a weak bisimulation relation.

Case 2. $K_1 \in S/R, K_2 = \cup_{C \in U} C$.

If $U = [C_0]_{\leftrightarrow}$ contains a sink C for U, i.e. for all $C' \in U$ we have $C' \to C$, we can write

$$s \xrightarrow{\tau^*\hat{a}\tau^*} C = s \xrightarrow{\tau^*\hat{a}\tau^*} {}_{\neg K_2} C \ \uplus \biguplus_{C' \in\, U - \{C\}} s \xrightarrow{\tau^*\hat{a}\tau^*} {}_{\neg K_2} C' \cdot C' \xrightarrow{\tau^*} C$$

and since there are at most countably many R-classes $C' \in U - \{C\}$ for which $s \xrightarrow{\tau^*\hat{a}\tau^*} {}_{\neg K_2} C' \neq \emptyset$, we get

$$\begin{aligned}
\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C) &= \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C, \neg K_2) \\
&\quad + \sum_{C' \in U - \{C\}} \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C', \neg K_2) \\
&= \sum_{C' \in U} \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C', \neg K_2) \\
&= \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, K_2).
\end{aligned}$$

The last equation holds since

$$s \xrightarrow{\tau^*\hat{a}\tau^*} K_2 = \biguplus_{C' \in\, U} s \xrightarrow{\tau^*\hat{a}\tau^*} {}_{\neg K_2} C'.$$

In the same way we get $\mathrm{Prob}(t, \tau^*\hat{a}\tau^*, C) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, K_2)$, thus

$$\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, K_2) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, K_2).$$

Note that we only used that U has a sink, and not that it is a whole class of the equivalence relation $\leftrightarrow$.

On the other hand, if U does not contain an R-class which is a sink (and this can only happen for infinite U because of Lemma 5.3.20), we use an approximation argument. Since there are at most countably many paths outgoing from s, there exists a countable set $U_s \subseteq U$ such that $\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{C \in U_s} C) = \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{C \in U} C)$. For the same reason, there exists $U_t \subseteq U$, a countable

set with the property $\mathrm{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{C\in U_t} C) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{C\in U} C)$. Taking $U' = U_s \cup U_t$ we get a countable set, such that both

$$\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{C\in U'} C) = \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, K_2) \tag{5.22}$$

and

$$\mathrm{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{C\in U'} C) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, K_2). \tag{5.23}$$

Let $\{C_i \mid i \in \mathbb{N}\}$ be an enumeration of U'. We will define a chain of subsets of U in the following way. Put $U_1 = \{C_1\}$ and

$$U_{n+1} = U_n \cup \{C_{n+1}\} \cup \{C^{n+1}\}$$

where $C^{n+1} \in S/R$ is a sink for $U_n \cup \{C_{n+1}\}$. Such a sink exists by Lemma 5.3.20, and it belongs to U, since U is a $\leftrightarrow$ equivalence class. We have $U_n \subseteq U_{n+1}$ for every natural number n, and also

$$U' \subseteq \bigcup_{n\in\mathbb{N}} U_n \subseteq U.$$

Next we denote some sets of finite paths. Let

$$\begin{aligned}
\Pi_s^n &= \{\pi \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) \in \tau^*\hat{a}\tau^*, \mathrm{last}(\pi) \in \cup_{C\in U_n} C\} \\
\Pi_s^U &= \{\pi \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) \in \tau^*\hat{a}\tau^*, \mathrm{last}(\pi) \in \cup_{C\in U} C\} \\
\Pi_s^{U'} &= \{\pi \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) \in \tau^*\hat{a}\tau^*, \mathrm{last}(\pi) \in \cup_{C\in U'} C\}
\end{aligned}$$

and similarly we use $\Pi_t^n, \Pi_t^U, \Pi_t^{U'}$. We have

$$\Pi_s^{U'} \subseteq \bigcup_{n\in\mathbb{N}} \Pi_s^n \subseteq \Pi_s^U$$

and similar holds for t in place of s. Furthermore, by (5.22) we have $\mathrm{Prob}(\Pi_s^{U'}) = \mathrm{Prob}(\Pi_s^U)$, hence

$$\mathrm{Prob}(\cup_{n\in\mathbb{N}} \Pi_s^n) = \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, K_2).$$

Also, by (5.23),

$$\mathrm{Prob}(\cup_{n\in\mathbb{N}} \Pi_t^n) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, K_2).$$

Now since $\Pi_s^n \subseteq \Pi_s^{n+1}$ and $\Pi_t^n \subseteq \Pi_t^{n+1}$ we get that

$$\begin{aligned}
\mathrm{Prob}(\cup_{n\in\mathbb{N}} \Pi_s^n) &= \lim_{n\to\infty} \mathrm{Prob}(\Pi_s^n) \\
&= \lim_{n\to\infty} \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{C\in U_n} C) \\
&\overset{(*)}{=} \lim_{n\to\infty} \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{C\in U_n} C) \\
&= \mathrm{Prob}(\cup_{n\in\mathbb{N}} \Pi_t^n)
\end{aligned}$$

where $(*)$ holds since each U_n is a set of R-classes that contains a sink, which completes the proof of this case.

Case 3. $K_1 = \cup_{C \in U} C,\ K_2 \in S/R'$

Consider $s, t \in K_1$. There exist R-classes C_1 and C_2 such that $s \in C_1$ and $t \in C_2$. We have $C_1 \leftrightarrow C_2$. By Lemma 5.3.19, there also exists an R-class C such that $C_1 \to C$ and $C_2 \to C$, and moreover $C \in U$, again since U is a $\leftrightarrow$ equivalence class.

If $K_2 = K_1$, then we have

$$\mathrm{Prob}(s, \tau^*, K_2) = \mathrm{Prob}(t, \tau^*, K_2) = 1.$$

If $K_2 \neq K_1$ then $K_2 \in S/R$ and $C \not\leftrightarrow K_2$. So, by Lemma 5.3.16 any τ^* path from C_i to K_2 must pass C, for $i \in \{1, 2\}$. Hence,

$$C_i \xrightarrow{\tau^*} K_2 \subseteq C_i \xrightarrow{\tau^*} C \cdot C \xrightarrow{\tau^*} K_2$$

and moreover, by Lemma 5.3.16, equality holds, i.e., $C_i \xrightarrow{\tau^*} C = C_i \xrightarrow{\tau^*}_{\neg(K_2 \cup C)} C$ since, if a τ^* path from C_i to C passes K_2 on the way, then either it was not minimal, i.e. it has a prefix that is also a τ^* path from C_i to C, or $K_2 \to C$ which is not possible, since $K_2 \neq K_1$. Hence, in this case

$$\begin{aligned}
\mathrm{Prob}(s, \tau^*, K_2) &= \mathrm{Prob}(s \xrightarrow{\tau^*} K_2) \\
&= \mathrm{Prob}(C_1, \tau^*, C) \cdot \mathrm{Prob}(C, \tau^*, K_2) \\
&= \mathrm{Prob}(C, \tau^*, K_2) \\
&= \mathrm{Prob}(C_2, \tau^*, C) \cdot \mathrm{Prob}(C, \tau^*, K_2) \\
&= \mathrm{Prob}(t, \tau^*, K_2).
\end{aligned}$$

Next we consider paths with traces in $\tau^* a \tau^*$. For $i \in \{1, 2\}$, and $K_2 \in S/R'$ arbitrary ($K_2 = K_1$ is also possible), by Lemma 5.3.17 we have

$$C_i \xrightarrow{\tau^* a \tau^*} K_2 \subseteq C_i \xrightarrow{\tau^*} C \cdot C \xrightarrow{\tau^* a \tau^*} K_2.$$

Here also equality holds, since no path on the right hand side can have a proper prefix in $C_i \xrightarrow{\tau^* a \tau^*} K_2$. Hence, similar as before,

$$\mathrm{Prob}(s, \tau^* a \tau^*, K_2) = \mathrm{Prob}(C, \tau^* a \tau^*, K_2) = \mathrm{Prob}(t, \tau^* a \tau^*, K_2).$$

The notation $\mathrm{Prob}(C, \tau^* a \tau^*, K_2)$ if $K_2 = K_1$ is justified by Case 2. □

We need one more property in order to prove Proposition 5.3.14.

Lemma 5.3.22. *Let R be a weak bisimulation on $\langle S, A, P \rangle$. Consider the set*

$$\mathcal{W} = \{R' \mid R' \text{ is a weak bisimulation on } \langle S, A, \mathrm{P} \rangle,\ R' \supseteq R\}$$

ordered by inclusion. Every chain of $\mathcal{W}$ has an upper bound.

Proof Let $\{R_i \mid i \in I\}$ be a chain of elements of $\mathcal{W}$, where I is also a chain of indices, and $R_i \subseteq R_j$ for $i \leq j$. We show that $\cup_{i\in I} R_i \in \mathcal{W}$. Note that if $C \in S/\cup_{i\in I} R_i$ is a class, then $C = \cup_{i\in I} C_i$ where $C_i \in S/R_i$, and $C_i \subseteq C_j$ for $i \leq j$.

The simplest case is when the chain has a largest element, say R_m and hence also $C = C_m$ and the property $\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C) = \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, C)$ for $\langle s,t\rangle \in \cup_{i\in I} R_i$ holds for R_m is a weak bisimulation.

We next treat the case when I is a countable set, ordered as the natural numbers, $I = \mathbb{N}$, i.e., $\{R_i \mid i \in \mathbb{N}\}$ is a countable chain, with $R_i \subseteq R_{i+1}$. Let $\langle s,t\rangle \in \cup_{i\in\mathbb{N}} R_i$. Then there exists j such that $\langle s,t\rangle \in R_j$, but also $\langle s,t\rangle \in R_n$ for all $n \geq j$. Consider the sets of paths

$$\Pi_s = \cup\{\pi\uparrow \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) = \tau^*\hat{a}\tau^*, \mathrm{last}(\pi) \in C\}$$
$$\Pi_s^i = \cup\{\pi\uparrow \mid \mathrm{first}(\pi) = s, \mathrm{trace}(\pi) = \tau^*\hat{a}\tau^*, \mathrm{last}(\pi) \in C_i\},\ i \in \mathbb{N}$$

Similarly, we use Π_t and Π_t^i. We have $\Pi_s = \cup_{i\in\mathbb{N}} \Pi_s^i$ and $\Pi_s^i \subseteq \Pi_s^{i+1}$ for all i. Hence,

$$\begin{aligned}
\mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C) &= \mathrm{Prob}(\Pi_s) \\
&= \mathrm{Prob}(\cup_{i\in\mathbb{N}} \Pi_s^i) \\
&\overset{(a)}{=} \lim_{n\to\infty} \mathrm{Prob}(\Pi_s^n) \\
&= \lim_{n\to\infty} \mathrm{Prob}(s, \tau^*\hat{a}\tau^*, C_n) \\
&\overset{(b)}{=} \lim_{n\to\infty} \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, C_n) \\
&= \mathrm{Prob}(t, \tau^*\hat{a}\tau^*, C)
\end{aligned}$$

where (a) holds since Prob is a measure, and (b) holds since for $n \geq j$ we have: $\langle s,t\rangle \in R_n$, C_n is an R_n-class, and R_n is a weak bisimulation.

We further show that if I is a countable chain of sets $\{C_i \mid i \in I\}$, then there exists a sub-chain I' of I with $\cup_{i\in I} C_i = \cup_{i\in I'} C_i$ and I' is either finite or isomorphic to ω, the order type of the natural numbers. We give the construction of I'. Given a countable chain I, denote by $f : \mathbb{N} \to I$ the bijection that exists since I is countable. Define a sequence of finite sub-chains of I by $I_0 = \{f(0)\}$ and

$$I_{n+1} = \begin{cases} I_n \cup \{f(n+1)\} & \forall i \in I_n : f(n+1) > i \\ I_n & \text{otherwise.} \end{cases}$$

Put

$$I' = \bigcup_{n\in\mathbb{N}} I_n.$$

It is straightforward to see that either I' is a finite chain, or I' is isomorphic to ω and in any case

$$\bigcup_{i\in I} C_i = \bigcup_{i\in I'} C_i.$$

Assume now that $\{R_i \mid i \in I\}$ is an arbitrary chain in $\mathcal{W}$. Let $\langle s,t\rangle \in \cup_{i\in I} R_i$, and let $C \in S/\cup_{i\in I} R_i$. Then $C = \cup_{i\in I} C_i$. Let

$$\Pi_s = \{\pi \mid \text{first}(\pi) = s, \text{trace}(\pi) = \tau^*\hat{a}\tau^*, \text{last}(\pi) \in C = \cup_{i\in I} C_i\}$$
$$\Pi_t = \{\pi \mid \text{first}(\pi) = t, \text{trace}(\pi) = \tau^*\hat{a}\tau^*, \text{last}(\pi) \in C = \cup_{i\in I} C_i\}$$

Let in be a function, $in : \Pi_s \cup \Pi_t \to I$ such that $\text{last}(\pi) \in C_{in(\pi)}$. Such a function exists by the definition of Π_s and Π_t. Then the set $I' = in(\Pi_s \cup \Pi_t) \subseteq I$ is at most countable since such are Π_s and Π_t. Furthermore, let

$$\Pi'_s = \{\pi \mid \text{first}(\pi) = s, \text{trace}(\pi) = \tau^*\hat{a}\tau^*, \text{last}(\pi) \in C = \cup_{i\in I'} C_i\}$$
$$\Pi'_t = \{\pi \mid \text{first}(\pi) = t, \text{trace}(\pi) = \tau^*\hat{a}\tau^*, \text{last}(\pi) \in C = \cup_{i\in I'} C_i\}$$

By the construction of I' we have that $\Pi_s = \Pi'_s$ and $\Pi_t = \Pi'_t$ and

$$\text{Prob}(s, \tau^*\hat{a}\tau^*, C) = \text{Prob}(\Pi_s) = \text{Prob}(\Pi'_s) \stackrel{(*)}{=} \text{Prob}(\Pi'_t) = \text{Prob}(t, \tau^*\hat{a}\tau^*, C).$$

The equality marked by $(*)$ holds since $\text{Prob}(\Pi'_s) = \text{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{i\in I'} C_i)$ and $\text{Prob}(\Pi'_t) = \text{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{i\in I'} C_i)$, and as proved above, in the case of a finite chain of classes or a countable chain of classes of order type ω, we have $\text{Prob}(s, \tau^*\hat{a}\tau^*, \cup_{i\in I'} C_i) = \text{Prob}(t, \tau^*\hat{a}\tau^*, \cup_{i\in I'} C_i)$. □

Finally, Proposition 5.3.14 follows from the lemmas 5.3.15-5.3.22.

Proof [of Proposition 5.3.14] By Lemma 5.3.22, since the set $\mathcal{W}$ is nonempty, as it contains R, and by Zorn's lemma we have that the set

$$\mathcal{W} = \{R' \mid R' \text{ is a weak bisimulation on } \langle S, A, \text{P}\rangle,\ R' \supseteq R\}$$

has a maximal element. Let it be $\tilde{R}$. Assume $\tilde{R}$ is not complete, i.e. there exists two different classes $C_1, C_2 \in S/\tilde{R}$ such that $C_1 \to C_2$. Then by Lemma 5.3.21 we can construct a weak bisimulation $\tilde{R}' \supset \tilde{R}$ which contradicts the maximality of $\tilde{R}$. Hence $\tilde{R}$ is complete i.e. for any two different $C_1, C_2 \in S/\tilde{R}$ we have $\text{Prob}(C_1, \tau^*, C_2) < 1$, and since $R \subseteq \tilde{R}$ it relates s and t which completes the proof. □

5.3.4 Weak coalgebraic bisimulation for generative systems

In this subsection we will provide a definition of weak bisimulation for generative systems, according to our coalgebraic approach from Section 5.1. For this we need a $*$-translation that will translate the generative systems with action set A to some systems with action set A^*. Unlike for the LTS, for the generative probabilistic systems, it does not seem possible to define a $*$-translation which will be of the same type, i.e. for the functor $\mathcal{G}_A$. Therefore, for coalgebraic weak probabilistic bisimulation, we shall consider yet another type of systems.

The functor

Let $\mathcal{G}^*$ be the bifunctor defined by

$$\mathcal{G}^*(A,S) = (\mathcal{P}(A) \times \mathcal{P}(S) \to [0,1])$$

on objects $\langle A, S\rangle$ and for morphisms $\langle f_1, f_2\rangle \colon A \times S \to B \times T$ by

$$\mathcal{G}^* f = (\nu \mapsto \nu \circ \langle f_1^{-1}, f_2^{-1}\rangle \mid \nu \colon \mathcal{P}(A) \times \mathcal{P}(S) \to [0,1]).$$

Consider the Set functor $\mathcal{G}^*_A$ corresponding to $\mathcal{G}^*$, so that

$$\mathcal{G}^*_A(S) = (\mathcal{P}(A) \times \mathcal{P}(S) \to [0,1])$$

and for a mapping $f \colon S \to T$,

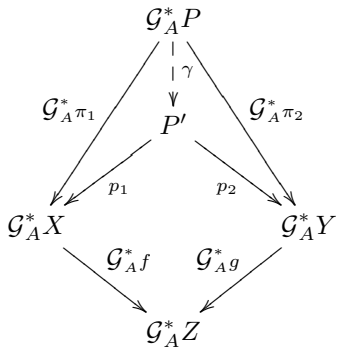
$$\mathcal{G}^*_A f = (\nu \mapsto \nu \circ \langle id_A^{-1}, f^{-1}\rangle \mid \nu \colon \mathcal{P}(A) \times \mathcal{P}(S) \to [0,1]).$$

Properties of the functor

We will use the functor $\mathcal{G}^*_A$ to model the $*$-translation of generative systems. Therefore we are interested in characterizing equivalence bisimulations for this functor. In order to apply the results from Section 3.6 we need the following.

Lemma 5.3.23. *The functor $\mathcal{G}^*_A$ weakly preserves total pullbacks.*

Proof Let $\langle P, \pi_1, \pi_2\rangle$ be a total Set pullback of the cospan $X \xrightarrow{f} Z \xleftarrow{g} Y$ i.e. $P = \{\langle x, y\rangle \mid f(x) = g(y)\}$ and π_1, π_2 surjective. Then the outer square of the following diagram commutes, and a morphism $\gamma : \mathcal{G}^*_A P \to P'$ exists, where $\langle P', p_1, p_2\rangle$ with p_1, p_2 projections, is the Set pullback of the cospan $\mathcal{G}^*_A X \xrightarrow{\mathcal{G}^*_A f} \mathcal{G}^*_A Z \xleftarrow{\mathcal{G}^*_A g} \mathcal{G}^*_A Y$.



According to Lemma 3.3.5, it is enough to prove that γ is surjective, i.e., that for every $\langle u, v\rangle \in P'$ there exists $w \in \mathcal{G}^*_A P$ with $\mathcal{G}^*_A \pi_1(w) = u$ and $\mathcal{G}^*_A \pi_2(w) = v$

which is equivalent to $w \circ \langle id_A^{-1}, \pi_1^{-1} \rangle = u$ and $w \circ \langle id_A^{-1}, \pi_2^{-1} \rangle = v$. Fix $\langle u, v \rangle \in P'$. We have

$$\langle u, v \rangle \in P' \implies \forall A' \subseteq A, \forall Z' \subseteq Z \colon u(A', f^{-1}(Z')) = v(A', g^{-1}(Z')) \quad (5.24)$$

Let $X' \subseteq X, Y' \subseteq Y$ and assume $\pi_1^{-1}(X') = \pi_2^{-1}(Y')$. Then

(i) $f^{-1}(f(X')) = X'$
Clearly $X' \subseteq f^{-1}(f(X'))$. Let $x \in f^{-1}(f(X'))$ such that $f(x) = f(x')$ for some $x' \in X'$. Since π_1 is surjective, there exists $y \in Y$ with $\langle x, y \rangle \in P$ i.e. $f(x) = g(y)$ and hence also $f(x') = g(y)$ i.e. $\langle x', y \rangle \in P$. Thus $\langle x', y \rangle \in \pi_1^{-1}(X') = \pi_2^{-1}(Y')$ implying $y \in Y'$. Hence $\langle x, y \rangle \in \pi_2^{-1}(Y') = \pi_1^{-1}(X')$ i.e. $x \in X'$.

(ii) $g^{-1}(g(Y')) = Y'$, similar as (i).

(iii) $f(X') = g(Y')$
Let $z \in f(X')$ i.e. $z = f(x')$ for $x' \in X'$. Since π_1 is surjective there exists $y \in Y$ with $\langle x', y \rangle \in P$ i.e. $f(x') = g(y)$. Now, $\langle x', y \rangle \in \pi_1^{-1}(X') = \pi_2^{-1}(Y')$ and therefore $y \in Y'$ i.e. $z = f(x') = g(y) \in g(Y')$. We have shown $f(X') \subseteq g(Y')$. Similarly, $g(Y') \subseteq f(X')$.

Hence, if $\pi_1^{-1}(X') = \pi_2^{-1}(Y')$ for $X' \subseteq X, Y' \subseteq Y$ we get, for any $A' \subseteq A$

$$u(A', X') \overset{(i)}{=} u(A', f^{-1}(f(X'))) \overset{(5.24)}{=} v(A', g^{-1}(f(X'))) \overset{(iii)}{=}$$

$$v(A', g^{-1}(g(Y'))) \overset{(ii)}{=} v(A', Y').$$

This, together with

$$\pi_1^{-1}(X') = \pi_1^{-1}(X'') \implies X' = X''$$

and

$$\pi_2^{-1}(Y') = \pi_2^{-1}(Y'') \implies Y' = Y''$$

for any $X', X'' \subseteq X$ and any $Y', Y'' \subseteq Y$ (π_1 and π_2 are surjective), shows that the function $w \colon \mathcal{P}(A) \times \mathcal{P}(P) \to [0, 1]$ given by

$$w(A', Q) = \begin{cases} u(A', X') & Q = \pi_1^{-1}(X') \\ v(A', Y') & Q = \pi_2^{-1}(Y') \\ 0 & \text{otherwise} \end{cases}$$

is well defined. Clearly, $w \circ \langle id_A^{-1}, \pi_1^{-1} \rangle = u$ and $w \circ \langle id_A^{-1}, \pi_2^{-1} \rangle = v$. Thus the functor $\mathcal{G}_A^*$ weakly preserves total pullbacks. □

Note that, however, $\mathcal{G}_A^*$ does not preserve weak pullbacks, as shown by the next example.

Example 5.3.24. $\mathcal{G}_A^*$ does not preserve weak pullbacks.

Choose X with $|X| \geq 3$. Fix $x_0 \in X$. Let $Z = \{1, 2, 3\}$ and consider the cospan $X \xrightarrow{f} Z \xleftarrow{g} X$ for the maps

$$f(x) = \begin{cases} 2 & x = x_0 \\ 1 & \text{otherwise} \end{cases} \qquad g(x) = \begin{cases} 2 & x = x_0 \\ 3 & \text{otherwise.} \end{cases}$$

The Set pullback of this cospan is then $P = \{\langle x_0, x_0\rangle\}$. On the other hand, let P' be the pullback of the cospan

$$\mathcal{G}_A^* X \xrightarrow{\mathcal{G}_A^* f} \mathcal{G}_A^* Z \xleftarrow{\mathcal{G}_A^* g} \mathcal{G}_A^* X \ .$$

We have $\langle \mu, \nu\rangle \in P'$ if and only if

$$\mathcal{G}_A^* f(\mu) = \mathcal{G}_A^* g(\nu)$$

i.e.

$$\mu(A', f^{-1}(X')) = \nu(A', g^{-1}(X'))$$

for all $A' \subseteq A, X' \subseteq X$. Therefore, every pair $\langle \mu, \nu\rangle \in \mathcal{G}_A^* X \times \mathcal{G}_A^* X$ with the property

$$\mu(A', \emptyset) = \mu(A', \{x_0\}) = \mu(A', X \setminus \{x_0\}) = \mu(A', X) =$$

$$= \nu(A', \emptyset) = \nu(A', \{x_0\}) = \nu(A', X \setminus \{x_0\}) = \nu(A', X)$$

belongs to P' since $\emptyset, \{x_0\}, X \setminus \{x_0\}$ and X are the only subsets of X that are inverse images of subsets of Z under f and g.

Now we consider $\mathcal{G}_A^* P = \{\chi \colon \mathcal{P}(A) \times \mathcal{P}(P) \to [0, 1]\}$. If $\mu \in \mathcal{G}_A^* X$ is such that $\mu = (\mathcal{G}_A^* \pi_1)(\chi)$ for some $\chi \in \mathcal{G}_A^* P$ then $\mu = \chi \circ \langle id_A^{-1}, \pi_1^{-1}\rangle$. Hence, for $A' \subseteq A, X' \subseteq X$ we have

$$\mu(A', X') = \begin{cases} \chi(A', \emptyset) & x_0 \notin X' \\ \chi(A', \{\langle x_0, x_0\rangle\}) & x_0 \in X'. \end{cases}$$

Choose $x_1 \in X$, $x_1 \neq x_0$. Since $|X| \geq 3$ we have $\{x_0, x_1\} \notin \{\emptyset, \{x_0\}, X \setminus \{x_0\}, X\}$. Define $\xi \colon \mathcal{P}(A) \times \mathcal{P}(X) \to [0, 1]$ by

$$\xi(A', X') = \begin{cases} 1 & X' = \{x_0, x_1\} \\ 0 & \text{otherwise.} \end{cases}$$

Then $\xi \in \mathcal{G}_A^*(X)$ and the pair $\langle \xi, \xi\rangle$ belongs to P' since for every $A' \subseteq A$,

$$\xi(A', \emptyset) = \xi(A', \{x_0\}) = \xi(A', X \setminus \{x_0\}) = \xi(A', X) = 0.$$

However, ξ can not be written as $(\mathcal{G}_A^* \pi_1)(\chi)$ for any $\chi \in \mathcal{G}_A^* P$ since

$$\xi(A', \{x_0, x_1\}) \neq \xi(A', \{x_0\}),$$

while, as noted above,

$$(\mathcal{G}_A^*\pi_1)(\chi)(A', \{x_0, x_1\}) = \chi(A', \{\langle x_0, x_0\rangle\}) = (\mathcal{G}_A^*\pi_1)(\chi)(A', \{x_0\}).$$

Hence, for the pair $\langle \xi, \xi\rangle \in P'$ there does not exist an element $\chi \in \mathcal{G}_A^* P$ such that $\mathcal{G}_A^*\pi_1(\chi) = \xi$ and $\mathcal{G}_A^*\pi_2(\chi) = \xi$, which by Lemma 3.3.5 shows that $\mathcal{G}_A^*$ does not preserve weak pullbacks.

Next, we investigate the bisimulations for the newly introduced type of systems.

Let R be an equivalence relation on a set S. A subset $M \subseteq S$ is an R-saturated set if for all $s \in M$ the whole equivalence class of s is contained in M. We denote by $\mathrm{Sat}(R)$ the set of all R-saturated sets, $\mathrm{Sat}(R) \subseteq \mathcal{P}(S)$. Actually, M is a saturated set if and only if $M = \cup_{i\in I} C_i$ for $C_i \in S/R$. Hence there is a one-to-one correspondence between the R-saturated sets and the elements of $\mathcal{P}(S/R)$.

The next lemma derives a transfer condition for equivalence bisimulations for systems of type $\mathcal{G}_A^*$.

Lemma 5.3.25. *An equivalence relation R on a set S is a bisimulation on the $\mathcal{G}_A^*$ system $\langle S, A, \alpha\rangle$ if and only if*

$$\langle s, t\rangle \in R \implies \forall A' \subseteq A, \forall M \in \mathrm{Sat}(R)\colon \alpha(s)(A', M) = \alpha(t)(A', M).$$

Proof Consider the pullback P of the cospan

$$\mathcal{G}_A^* S \xrightarrow{\mathcal{G}_A^* c} \mathcal{G}_A^*(S/R) \xleftarrow{\mathcal{G}_A^* c} \mathcal{G}_A^* S$$

where c is the canonical projection of S onto S/R. We have $\langle \mu, \nu\rangle \in P$ if and only if $\mathcal{G}_A^* c(\mu) = \mathcal{G}_A^* c(\nu)$, i.e. $\mu \circ \langle id_A^{-1}, c^{-1}\rangle = \nu \circ \langle id_A^{-1}, c^{-1}\rangle$. This is equivalent to

$$\forall A' \subseteq A, \forall M \subseteq S/R\colon \mu(A', c^{-1}(M)) = \nu(A', c^{-1}(M))$$

and, since $c^{-1} : \mathcal{P}(S/R) \to \mathrm{Sat}(R)$ is a bijection, we get an equivalent condition

$$\forall A' \subseteq A, \forall M \in \mathrm{Sat}(R)\colon \mu(A', M) = \nu(A', M).$$

Now, using Corollary 3.6.6 we obtain the stated characterization. □

We proceed by presenting a suitable $*$-translation for generative systems.

The $*$-translation

We can now define a $*$-translation for generative systems. The translation will be of type $\mathcal{G}_A^*$.

Definition 5.3.26. *Let Φ^g assign to every generative system $\langle S, A, \mathrm{P}\rangle$, i.e. any $\mathcal{G}_A$ coalgebra $\langle S, A, \alpha\rangle$ the $\mathcal{G}_{A^*}^*$ coalgebra $\langle S, A^*, \alpha'\rangle$ where for $W \subseteq A^*$ and $S' \subseteq S$, $\alpha'(s)(W, S') = \mathrm{Prob}(s, W, S')$.*

We next show that the defined translation is indeed a $*$-translation.

Theorem 5.3.27. *The assignment Φ^g from Definition 5.3.26 is a $*$-translation.*

For the proof we need an auxiliary property.

Lemma 5.3.28. *Let $\langle S, A, \alpha\rangle$, i.e. $\langle S, A, \mathrm{P}\rangle$ be a $\mathcal{G}_A$ system, R a bisimulation equivalence on $\langle S, A, \alpha\rangle$ and $\langle s, t\rangle \in R$. For $k \in \mathbb{N}, C_i \in S/R$ and $a_i \in A$, $i \in \{1, \ldots, k\}$, let $s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k$ denote the set of paths*

$$s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k = \{s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_k} s_k \mid s_i \in C_i, i = 1, \ldots, k\}.$$

Then $s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k$ is minimal and

$$\mathrm{Prob}(s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k) = \mathrm{Prob}(t \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k) \quad (5.25)$$

Proof The fact that $s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k$ is minimal is clear, since all paths in this set have the same length. We use induction on k to establish (5.25). For $k = 1$ the statement is $\sum_{s' \in C_1} \mathrm{P}(s, a_1, s') = \sum_{s' \in C_1} \mathrm{P}(t, a_1, s')$ and it holds since R is a bisimulation relation and $\langle s, t\rangle \in R$. Consider

$$s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_{k+1}} C_{k+1} = s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k \cdot C_k \xrightarrow{a_{k+1}} C_{k+1}.$$

By the inductive hypothesis,

$$\mathrm{Prob}(s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k) = \mathrm{Prob}(t \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k).$$

By the bisimulation condition for generative systems, $\mathrm{Prob}(t' \xrightarrow{a_{k+1}} C_{k+1}) = \mathrm{Prob}(t'' \xrightarrow{a_{k+1}} C_{k+1})$ for all $t', t'' \in C_k$. Hence, by Corollary 5.3.10 we get

$$\begin{aligned}
&\mathrm{Prob}(s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k \cdot C_k \xrightarrow{a_{k+1}} C_{k+1}) \\
&\quad = \mathrm{Prob}(s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k) \cdot \mathrm{Prob}(C_k \xrightarrow{a_{k+1}} C_{k+1}) \\
&\quad = \mathrm{Prob}(t \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k) \cdot \mathrm{Prob}(C_k \xrightarrow{a_{k+1}} C_{k+1}) \\
&\quad = \mathrm{Prob}(t \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k \cdot C_k \xrightarrow{a_{k+1}} C_{k+1}).
\end{aligned}$$

□

We are now prepared for the proof of Theorem 5.3.27.

Proof [of Theorem 5.3.27] We need to check that Φ^g is injective and preserves and reflects bisimilarity. Assume $\Phi^g(\langle S, A, \alpha\rangle) = \Phi^g(\langle S, A, \beta\rangle) = \langle S, A^*, \alpha'\rangle$. Then by the definition of Prob we get that for any $s, t \in S$ and any $a \in A$, $\alpha(s)(\langle a, t\rangle) = \mathrm{P}(s, a, t) = \mathrm{Prob}(s, \{a\}, \{t\}) = \alpha'(s)(\{a\}, \{t\}) = \beta(s)(\langle a, t\rangle)$.

Reflection of bisimilarity is direct from Lemma 5.3.25: Assume $s \sim t$ in $\Phi^g(\langle S, A, \alpha\rangle) = \langle S, A^*, \alpha'\rangle$ and assume that R is an equivalence bisimulation

on $\langle S, A^*, \alpha' \rangle$ such that $\langle s, t \rangle \in R$. By Lemma 5.3.25, we get that for $W \subseteq A^*$ and for $M \in \mathrm{Sat}(R)$,

$$\alpha'(s)(W, M) = \alpha'(t)(W, M). \tag{5.26}$$

In particular, for all $a \in A$ and all $C \in S/R$ we have

$$\alpha'(s)(\{a\}, C) = \alpha'(t)(\{a\}, C). \tag{5.27}$$

By the definition of α' and Prob we have

$$\alpha'(s)(\{a\}, C) = \mathrm{Prob}(s, \{a\}, C) = \sum_{s' \in C} \mathrm{P}(s, a, s') = \sum_{s' \in C} \alpha(s)(\langle a, s' \rangle)$$

and therefore for all $a \in A$ and all $C \in S/R$

$$\sum_{s' \in C} \alpha(s)(\langle a, s' \rangle) = \sum_{s' \in C} \alpha(t)(\langle a, s' \rangle) \tag{5.28}$$

which means that R is a bisimulation equivalence on the generative system $\langle S, A, \alpha \rangle$, i.e. $s \sim t$ in the original system.

The proof of preservation of bisimilarity uses Lemma 5.3.28. Let $s \sim t$ in the generative system $\langle S, A, \alpha \rangle$. Then there exists an equivalence bisimulation R with $\langle s, t \rangle \in R$. The relation R induces an equivalence R_P on $\mathrm{FPaths}(s)$ defined by

$$\langle s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_k} s_k \ , \ s \xrightarrow{a'_1} s'_1 \xrightarrow{a'_2} s'_2 \cdots \xrightarrow{a'_{k'}} s'_{k'} \rangle \in R_P$$

if and only if $k = k'$, $a_i = a'_i$ and $\langle s_i, s'_i \rangle \in R$ for $i = 1, \ldots, k$. The classes of R_P are exactly the sets $s \xrightarrow{a_1} C_1 \xrightarrow{a_2} C_2 \cdots \xrightarrow{a_k} C_k$ for $C_i \in S/R$ and $a_i \in A$.

Assume $M \in \mathrm{Sat}(R)$ and $W \subseteq A^*$. We show that the set $s \xrightarrow{W} M$ is saturated with respect to R_P. Namely, let $\pi \equiv s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_k} s_k \in s \xrightarrow{W} M$ and let $\pi' \equiv s \xrightarrow{a_1} s'_1 \xrightarrow{a_2} s'_2 \cdots \xrightarrow{a_k} s'_k$ be a path such that $\langle \pi, \pi' \rangle \in R_P$. Then $\mathrm{trace}(\pi) = \mathrm{trace}(\pi')$, $\mathrm{first}(\pi) = \mathrm{first}(\pi')$ and $\langle \mathrm{last}(\pi), \mathrm{last}(\pi') \rangle \in R$. Since M is saturated, $\mathrm{last}(\pi') \in M$ for $\mathrm{last}(\pi) \in M$. Furthermore, π' does not have a proper prefix with trace in W and last in M, since this would imply that π has such a prefix, contradicting $\pi \in s \xrightarrow{W} M$. Hence, $\pi' \in s \xrightarrow{W} M$.

Therefore, the set $s \xrightarrow{W} M$ is a disjoint union of some R_P classes, and since $s \xrightarrow{W}\!\!\!\!\!\rightarrow M$ is minimal we can write

$$s \xrightarrow{W}\!\!\!\!\!\rightarrow M = \biguplus_{i \in I} s \xrightarrow{a_{i1}}\!\!\!\!\!\rightarrow C_{i1} \xrightarrow{a_{i2}}\!\!\!\!\!\rightarrow C_{i2} \cdots \xrightarrow{a_{ik_i}}\!\!\!\!\!\rightarrow C_{ik_i} \, .$$

It follows that $\mathrm{Prob}(s, W, M) = \sum_{i \in I} \mathrm{Prob}(s \xrightarrow{a_{i1}} C_{i1} \xrightarrow{a_{i2}} C_{i2} \cdots \xrightarrow{a_{ik}} C_{ik})$. By Lemma 5.3.28, we get that $\mathrm{Prob}(s, W, M) = \mathrm{Prob}(t, W, M)$, i.e. $\alpha'(s)(W, M) = \alpha'(t)(W, M)$ proving that R is a bisimulation on $\langle S, A^*, \alpha' \rangle$ and $s \sim t$ in the *-extension $\langle S, A^*, \alpha' \rangle$. □

The same systems of Example 5.2.4 when each transition is considered as probabilistic with probability 1 show that the $*$-translation Φ^g is also not induced by a natural transformation.

Remark 5.3.29. The $*$-translation Φ^g together with a subset $\tau \subseteq A$ determines a weak-τ-bisimulation. The weak-τ-system is

$$\Psi_\tau \circ \Phi^g(\langle S, A, \alpha\rangle) = \Psi_\tau(\langle S, A^*, \alpha'\rangle) = \langle S, A_\tau, \alpha''\rangle$$

where $\alpha''(s) : \mathcal{P}(A_\tau) \times \mathcal{P}(S) \to [0,1]$ is given by

$$\alpha''(s) = \ \eta^\tau_S(\alpha'(s)) = \ \mathcal{G}^*\langle h_\tau, id_S\rangle(\alpha'(s)) = \ \alpha'(s) \circ \langle h_\tau^{-1}, id_S^{-1}\rangle.$$

Hence for $X \subseteq A_\tau$ and $S' \subseteq S$,

$$\alpha''(s)(X, S') = \alpha'(s)(h_\tau^{-1}(X), S') = \alpha'(s)(\bigcup_{w\in X} B_w, S') = \mathrm{Prob}(s, \bigcup_{w\in X} B_w, S'),$$

where, as before, for $w = a_1 \dots a_k \in A_\tau$, B_w is the block $B_w = \tau^* a_1 \tau^* \dots \tau^* a_k \tau^* = h_\tau^{-1}(\{w\})$.

Therefore, from Lemma 5.3.25 we get that an equivalence relation R is a weak-τ-bisimulation w.r.t. $\langle \Phi^g, \tau\rangle$ on the generative system $\langle S, A, \alpha\rangle$ if and only if $\langle s, t\rangle \in R$ implies that for any collection $(B_i)_{i\in I}$ of blocks writing B_i as a shorthand for B_{w_i} for some word $w_i \in A^*$, and any collection $(C_j)_{j\in J}$ of classes

$$\mathrm{Prob}(s, \bigcup_{i\in I} B_i, \bigcup_{j\in J} C_j) = \mathrm{Prob}(t, \bigcup_{i\in I} B_i, \bigcup_{j\in J} C_j). \tag{5.29}$$

Sets of the form $\cup_{i\in I} B_i$ will be called *saturated blocks*.

5.3.5 The correspondence theorem

We are now able to state and prove the correspondence theorem.

Theorem 5.3.30. *Let $\langle S, A, \alpha\rangle$ be a generative system. Let $\tau \in A$ be an invisible action and $s, t \in S$ any two states. Then $s \approx_{\{\tau\}} t$ according to Definition 5.1.3 w.r.t. the pair $\langle \Phi^g, \{\tau\}\rangle$ if and only if $s \approx_g t$ according to Definition 5.3.12.*

In order to build the proof of the necessity part of the correspondence theorem, we present a sequence of lemmas.

Lemma 5.3.31. *Assume that R is a complete weak bisimulation on a generative system $\langle S, A, \alpha\rangle$ i.e. $\langle S, A, \mathrm{P}\rangle$ with $\langle s, t\rangle \in R$. For any saturated set $M = \sqcup_{i=1}^n C_i$ consisting of finitely many classes $C_i \in S/R$, for any block $B = \tau^* a_1 \tau^* \dots \tau^* a_k \tau^*$ where $a_1, \dots, a_k \in A \setminus \{\tau\}$ and for any $i \in \{1, \dots, n\}$,*

$$\mathrm{Prob}(s, B, C_i, \neg M) = \mathrm{Prob}(t, B, C_i, \neg M).$$

Proof We use induction on n, the number of classes that M contains. For $n = 1$ the property is simply Proposition 5.3.13. Assume $\mathrm{Prob}(s, B, C_i, \neg M) = \mathrm{Prob}(t, B, C_i, \neg M)$ for any R - saturated set M being a union of less than n classes, and any class $C_i \subseteq M$. Let M be an R- saturated set which is a union of n classes, i.e. $M = \sqcup_{i=1}^{n} C_i$ for some $C_i \in S/R$. We use the following notation, for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$.

$$V_i = \mathrm{Prob}(s, B, C_i) \stackrel{5.3.13}{=} \mathrm{Prob}(t, B, C_i)$$

$$G_i^j = \mathrm{Prob}(s, B, C_j, \neg \sqcup_{k=1, k\neq i}^{n} C_k) \stackrel{IH}{=} \mathrm{Prob}(t, B, C_j, \neg \sqcup_{k=1, k\neq i}^{n} C_k)$$

which makes the following notation justified

$$T_i^j = \mathrm{Prob}(C_j, \tau^*, C_i)$$

$$H_i^j = \mathrm{Prob}(C_i, \tau^*, C_j, \neg \sqcup_{k=1, k\neq i}^{n} C_k).$$

We define a function $\omega : s \stackrel{B}{\rightarrow} S \rightarrow \{1,2\}^*$. The function ω will, in a sense, trace the classes that a path visits with a word in B. Some auxiliary functions will be needed for the definition of ω. Let $\tilde{\omega} : s \stackrel{B}{\rightarrow} S \rightarrow \{1,2\}^*$ be defined by

$$\tilde{\omega}(\pi \cdot \mathrm{last}(\pi) \stackrel{a}{\rightarrow} t) = \begin{cases} 1 & t \in C_i, \pi \notin s \stackrel{B}{\rightarrow} S \\ 2 & t \in M \setminus C_i, \pi \notin s \stackrel{B}{\rightarrow} S \\ \varepsilon & t \notin M, \pi \notin s \stackrel{B}{\rightarrow} S \\ \tilde{\omega}(\pi) \cdot 1 & t \in C_i, \pi \in s \stackrel{B}{\rightarrow} S \\ \tilde{\omega}(\pi) \cdot 2 & t \in M \setminus C_i, \pi \in s \stackrel{B}{\rightarrow} S \\ \tilde{\omega}(\pi) & t \notin M, \pi \in s \stackrel{B}{\rightarrow} S \end{cases}$$

and if $\varepsilon \in s \stackrel{B}{\rightarrow} S$, then $\tilde{\omega}(\varepsilon) = \varepsilon$.

Let $d : \{1,2\}^* \rightarrow \{1,2\}^*$ and $d' : \{1,2\}^* \rightarrow \{1,2\}^*$ be defined in the following way, for $u, v \in \{1,2\}^*$ and $x, y \in \{1,2\}$:

$$d(u \cdot x) = \begin{cases} d(u) \cdot x & u = v \cdot x \\ d'(u) \cdot x & u = v \cdot y, y \neq x \end{cases}$$

$$d'(u \cdot x) = \begin{cases} d'(u) & u = v \cdot x \\ d'(u) \cdot x & u = v \cdot y, y \neq x \end{cases}$$

and $d(\varepsilon) = d'(\varepsilon) = \varepsilon$. We put $\omega = d \circ \tilde{\omega}$. We can explain the definition of the maps d, $\tilde{\omega}$ and ω as follows: The map $\tilde{\omega}$ takes a path with a trace in B and encodes the sequence of the classes that are visited by the path, after a word in B has already been performed. The encoding is 1 if the class under consideration,

C_i, has been visited and 2 if any other class from M has been visited, there is no record of classes outside M. Then the map d removes adjacent multiple occurrences of 1 and 2 in the word obtained by $\tilde{\omega}$, except for the multiple occurrences at the end of the word. Basically, the map d is computed by the normal algorithm $\{112 \to 12, 221 \to 21\}$. It is important to note that

$$\omega^{-1}(\{1\}) = \ s \xrightarrow{B}_{\neg M} C_i$$

hence, we need to calculate $\mathrm{Prob}(\omega^{-1}(\{1\}))$. By the definition of ω we easily get that

$$\omega^{-1}(\{1, 21\}) = \omega^{-1}(\{1\}) \uplus \omega^{-1}(\{21\}).$$

Therefore, we try to express $\mathrm{Prob}(\omega^{-1}(\{1, 21\}))$ and $\mathrm{Prob}(\omega^{-1}(\{21\}))$ via V_i, G_i^j, T_i^j and H_i^j. It is obvious that

$$\omega^{-1}(\{1, 21\}) = \ s \xrightarrow{B} C_i,$$

and therefore $\mathrm{Prob}(\omega^{-1}(\{1, 21\})) = V_i$. A more careful inspection shows that

$$\omega^{-1}(\{21\}) \uplus \left(\uplus_{j=1, j\neq i}^{n} \omega^{-1}(\{1\}) \cdot C_i \xrightarrow{\tau^*}_{\neg M \setminus C_i} C_j \xrightarrow{\tau^*} C_i \right)$$

$$= \uplus_{j=1, j\neq i}^{n} s \xrightarrow{B}_{\neg M \setminus C_i} C_j \xrightarrow{\tau^*} C_i.$$

This, by Proposition 5.3.9 and Corollary 5.3.10 implies that

$$\mathrm{Prob}(\omega^{-1}(\{21\})) = \sum_{j=1, j\neq i}^{n} G_i^j \cdot T_i^j - \mathrm{Prob}(\omega^{-1}(\{1\})) \cdot \sum_{j=1, j\neq i}^{n} H_i^j \cdot T_i^j$$

and we get

$$\begin{aligned} \mathrm{Prob}(\omega^{-1}(\{1\})) &= \mathrm{Prob}(\omega^{-1}(\{1, 21\})) - \mathrm{Prob}(\omega^{-1}(\{21\})) \\ &= V_i - \left(\sum_{j=1, j\neq i}^{n} G_i^j \cdot T_i^j - \mathrm{Prob}(\omega^{-1}(\{1\})) \cdot \sum_{j=1, j\neq i}^{n} H_i^j \cdot T_i^j \right). \end{aligned}$$

Let $\rho = \sum_{j=1, j\neq i}^{n} H_i^j \cdot T_i^j$. Let $T_i = \max_{j=1, j\neq i}^{n} T_i^j$. By the completeness of R, $T_i^j < 1$ for all $j \neq i$ and therefore $T_i < 1$. Furthermore, by Proposition 5.3.11,

$$\sum_{j=1, j\neq i}^{n} H_i^j = \mathrm{Prob}(C_i, \tau^*, \sqcup_{j=1, j\neq i}^{n} C_j) \leq 1.$$

Hence,

$$\rho = \sum_{j=1, j\neq i}^{n} H_i^j \cdot T_i^j \leq T_i \cdot \sum_{j=1, j\neg i} H_i^j \leq T_i < 1.$$

We have

$$\mathrm{Prob}(s, B, C_i, \neg M) = V_i - \sum_{j=1, j\neq i}^{n} G_i^j \cdot T_i^j + \mathrm{Prob}(s, B, C_i, \neg M) \cdot \rho.$$

and since $\rho < 1$ we obtain

$$\mathrm{Prob}(s, B, C_i, \neg M) = \frac{V_i - \sum_{j=1, j\neq i}^{n} G_i^j \cdot T_i^j}{1 - \rho}.$$

The expression on the right side does not depend on the starting state s and we get

$$\mathrm{Prob}(s, B, C_i, \neg M) = \mathrm{Prob}(t, B, C_i, \neg M)$$

which completes the proof. □

Next we extend the property to arbitrary R-saturated sets.

Lemma 5.3.32. *Assume that R is a complete weak bisimulation on a generative system $\langle S, A, \alpha\rangle$ i.e. $\langle S, A, \mathrm{P}\rangle$ with $\langle s, t\rangle \in R$. For any R-saturated set M, for any block $B = \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^*$ where $a_1, \ldots, a_k \in A \setminus \{\tau\}$ and for any class $C \subseteq M$*

$$\mathrm{Prob}(s, B, C, \neg M) = \mathrm{Prob}(t, B, C, \neg M).$$

Proof Let $C \subseteq M$. We will show that we can assume that M contains at most countably many classes. Let S' be the set of states that are reachable from s by a finite path. This set is at most countable since each finite path contributes to S' with finitely many states, and there are at most countably many paths starting in s according to Lemma 5.3.2. Let M_s be the smallest R-saturated set containing $S' \cap M$ and C. Since $S' \cap M$ is at most countable, the set M_s contains at most countably many classes and $\mathrm{Prob}(s, B, C, \neg M) = \mathrm{Prob}(s, B, C, \neg M_s)$. In the same way we get a saturated set M_t containing at most countably many classes such that $\mathrm{Prob}(t, B, C, \neg M) = \mathrm{Prob}(t, B, C, \neg M_t)$. Then $M' = M_s \cup M_t$ is a saturated set containing at most countably many classes and

$$\mathrm{Prob}(s, B, C, \neg M') = \mathrm{Prob}(s, B, C, \neg M),$$

$$\mathrm{Prob}(t, B, C, \neg M') = \mathrm{Prob}(t, B, C, \neg M).$$

So, assume $M = \sqcup_{i\geq 0} C_i$, and $C = C_{i_0}$. Note that

$$s \xrightarrow{B}_{\neg M} C = \bigcap_{k \geq i_0} s \xrightarrow{B}_{\neg U_k} C$$

for $U_k = C_0 \cup \cdots \cup C_k$, and the intersection is clearly countable. Moreover, let

$$J = \{I \mid I \subseteq \mathbb{N} \setminus \{0, \ldots, i_0 - 1\}, I \text{ is finite}\}.$$

If $I \in J$ with $m = \max(I)$, then

$$\bigcap_{i \in I} s \xrightarrow{B}_{\neg U_i} C = s \xrightarrow{B}_{\neg U_m} C.$$

We use the following simple property from measure theory: If μ is a probability measure on some set and if $A = \cap_{n \in \mathbb{N}} A_n$ is a measurable set which is a countable intersection of measurable sets, then $\mu(A) = \inf\{\mu(\cap_{i \in I} A_i) \mid I \subseteq \mathbb{N}, I \text{ finite } \}$. Hence,

$$\begin{aligned}
&\mathrm{Prob}(s, B, C, \neg M) \\
&\quad = \inf\{\mathrm{Prob}(\cap_{i \in I} s \xrightarrow{B}_{\neg U_i} C) \mid I \in J\} \\
&\quad = \inf\{\mathrm{Prob}(s, B, C, \neg U_m) \mid I \ \in J, m = \max(I)\} \\
&\quad \overset{5.3.31}{=} \inf\{\mathrm{Prob}(t, B, C, \neg U_m) \mid I \in J, m = \max(I)\} \\
&\quad = \mathrm{Prob}(t, B, C, \neg M)
\end{aligned}$$

□

By Lemma 5.3.32, noting that $\mathrm{Prob}(s, B, M) = \mathrm{Prob}(s, B, \sqcup_{i \in I} C_i) = \sum_{i \in I} \mathrm{Prob}(s, B, C_i, \neg M)$ we get the following property.

Corollary 5.3.33. *Assume that R is a complete weak bisimulation on a generative system $\langle S, A, \alpha \rangle$ i.e. $\langle S, A, \mathrm{P} \rangle$ with $\langle s, t \rangle \in R$. For any R-saturated set M, for any block $B = \tau^* a_1 \tau^* \ldots \tau^* a_k \tau^*$ where $a_1, \ldots, a_k \in A \setminus \{\tau\}$*

$$\mathrm{Prob}(s, B, M) = \mathrm{Prob}(t, B, M).$$

□

We proceed to saturated blocks. Again we first treat saturated blocks containing finitely many blocks and then extend to arbitrary saturated blocks.

Lemma 5.3.34. *Assume that R is a complete weak bisimulation on a generative system $\langle S, A, \alpha \rangle$ i.e. $\langle S, A, \mathrm{P} \rangle$ with $\langle s, t \rangle \in R$. For any R-saturated set M and for any saturated block $W = \sqcup_{i=1}^{n} B_i$ containing finitely many blocks*

$$\mathrm{Prob}(s, W, M) = \mathrm{Prob}(t, W, M).$$

Proof Note that

$$\mathrm{Prob}(s, W, M) = \sum_{i=1}^{n} \mathrm{Prob}(s, B_i, \neg W, M)$$

since

$$s \xrightarrow{W} M = \biguplus_{i=1}^{n} s \xrightarrow{B_i}_{\neg W} M,$$

and also

$$\mathrm{Prob}(s, B_i, \neg W, M) = \sum_{j:\ C_j \subseteq M} \mathrm{Prob}(s, B_i, \neg W, C_j, \neg M)$$

since

$$s \xrightarrow{B_i}_{\neg W} M = \biguplus_{C_j \subseteq M} s \xrightarrow[\neg M]{B_i}_{\neg W} C_j,$$

as in Proposition 5.3.11, and the summation is possible since we can assume that M contains at most countably many classes. Hence it suffices to prove that

$$\mathrm{Prob}(s, B_i, \neg W, C_j, \neg M) = \mathrm{Prob}(t, B_i, \neg W, C_j, \neg M)$$

for any B_i, $i \in \{1, \ldots, n\}$ and any class $C_j \subseteq M$. For any i, let $w_i \in A \setminus \{\tau\}^*$, $w_i = a_{i1} \ldots a_{ik_i}$ be the word such that $B_i = B_{w_i} = \tau^* a_{i1} \tau^* \cdots \tau^* a_{ik_i} \tau^*$. The prefix ordering on the set of words $\{w_1, \ldots, w_n\}$ induces an ordering on the set of blocks $\{B_1, \ldots, B_n\}$ given by $B_i \prec B_j$ if and only if $w_i \prec w_j$. If $B_i \prec B_j$, by B_{j-i} we denote the block corresponding to w_{j-i}, the unique word satisfying $w_i \cdot w_{j-i} = w_j$. We are going to prove, by induction on the number of elements in the set $\{i \in \{1, \ldots, n\} \mid B_i \prec B_j\}$, that

$$s \xrightarrow{B_j}_{\neg M} C = s \xrightarrow[\neg M]{B_j}_{\neg W} C \uplus \left(\biguplus_{B_i \prec B_j} \biguplus_{C' \subseteq M} s \xrightarrow[\neg M]{B_i}_{\neg W} C' \xrightarrow{B_{j-i}}_{\neg M} C \right) \tag{5.30}$$

where $C' \subseteq M$ is a class. First of all we have to make sure that the right hand side of the equation is well defined, i.e. that the unions are really disjoint and minimal. By the definition of the involved sets of paths a careful inspection shows that it is indeed the case. It is rather obvious that the right hand side is contained in the left hand side since all the paths of the right hand side do start in s, have a trace in B_j and end up in C, without reaching M before with a prefix whose trace is also in B_j. For the opposite inclusion we use an inductive argument. Assume B_j has no (strict) prefixes in $\{B_1, \ldots, B_n\}$. Then the equation becomes $s \xrightarrow{B_j}_{\neg M} C = s \xrightarrow[\neg M]{B_j}_{\neg W} C$ and it holds since, by assumption, no path which has a trace in B_j can have a strict prefix with a trace in W. For the inductive step, assume $\pi \in s \xrightarrow{B_j}_{\neg M} C$ and $\pi \notin s \xrightarrow[\neg M]{B_j}_{\neg W} C$. This means that π has a prefix that has a trace in $\cup_{i=1}^{n} B_i$ and ends in M. So, $\pi \in s \xrightarrow{B_k} C' \xrightarrow{B_{j-k}}_{\neg M} C$ for some k and for some class $C' \subseteq M$. We want to show that $\pi \in \uplus_{B_i \prec B_j} \uplus_{C' \subseteq M} s \xrightarrow[\neg M]{B_i}_{\neg W} C' \xrightarrow{B_{j-i}}_{\neg M} C$. We can assume that $\pi \in s \xrightarrow{B_k}_{\neg M} C' \xrightarrow{B_{j-k}}_{\neg M} C$ by taking C' to be the first class of M that π hits having performed a trace in B_k. Now B_k, being a prefix of B_j, has less prefixes than B_j and therefore, by the inductive hypothesis, either

$$\pi \in s \xrightarrow[\neg M]{B_k}_{\neg W} C' \xrightarrow{B_{j-k}}_{\neg M} C$$

or there exist $r \in \{1, \ldots, n\}$ and a class $C'' \subseteq M$ such that

$$\pi \in s \xrightarrow[\neg M]{B_r, \neg W} C'' \xrightarrow{B_{k-r}, \neg M} C' \xrightarrow{B_{j-k}, \neg M} C$$

i.e. $\pi \in s \xrightarrow[\neg M]{B_r, \neg W} C'' \xrightarrow{B_{j-r}, \neg M} C$, which completes the proof of equation (5.30). Now, by the same inductive argument we get: if B_j has no proper prefixes, then

$$\begin{aligned} \mathrm{Prob}(s, B_j, \neg W, C, \neg M) &= \mathrm{Prob}(s, B_j, C, \neg M) \\ &\overset{5.3.32}{=} \mathrm{Prob}(t, B_j, C, \neg M) \\ &= \mathrm{Prob}(t, B_j, \neg W, C, \neg M). \end{aligned}$$

Assume that $\mathrm{Prob}(s, B_i, \neg W, C, \neg M) = \mathrm{Prob}(t, B_i, \neg W, C, \neg M)$ for all $B_i \prec B_j$. Then by (5.30), by Proposition 5.3.9 and by Lemma 5.3.32, we get

$$\begin{aligned} &\mathrm{Prob}(s, B_j, \neg W, C, \neg M) \\ &\quad = \mathrm{Prob}(s, B_j, C, \neg M) \\ &\qquad - \sum_{B_i \prec B_j} \sum_{C' \subseteq M} \mathrm{Prob}(s, B_i, \neg W, C', \neg M) \cdot \mathrm{Prob}(C', B_{j-i}, C, \neg M) \\ &\quad \overset{(IH)}{=} \mathrm{Prob}(t, B_j, C, \neg M) \\ &\qquad - \sum_{B_i \prec B_j} \sum_{C' \subseteq M} \mathrm{Prob}(t, B_i, \neg W, C', \neg M) \cdot \mathrm{Prob}(C', B_{j-i}, C, \neg M) \\ &\quad = \mathrm{Prob}(t, B_j, \neg W, C, \neg M) \end{aligned}$$

which completes the proof. □

We next extend the last property to any saturated block.

Lemma 5.3.35. *Assume that R is a complete weak bisimulation on a generative system $\langle S, A, \alpha \rangle$ i.e. $\langle S, A, \mathrm{P} \rangle$ with $\langle s, t \rangle \in R$. For any R-saturated set M and for any saturated block W*

$$\mathrm{Prob}(s, W, M) = \mathrm{Prob}(t, W, M).$$

Proof We first consider the countable case. Let $W = \sqcup_{n \in \mathbb{N}} B_n$. Let

$$\Pi_n^s = \{\pi \mid \mathrm{first}(\pi) = s, \mathrm{last}(\pi) \in M, \mathrm{trace}(\pi) \in B_n\}$$

$$\Pi_n^t = \{\pi \mid \mathrm{first}(\pi) = t, \mathrm{last}(\pi) \in M, \mathrm{trace}(\pi) \in B_n\}.$$

Then

$$\begin{aligned}
\mathrm{Prob}(s, W, M) &= \mathrm{Prob}(s, \sqcup_{n\in\mathbb{N}} B_n, M) \\
&= \mathrm{Prob}((\cup_{n\in\mathbb{N}} \Pi^s_n)\downarrow) \\
&= \mathrm{Prob}(\cup_{n\in\mathbb{N}} \Pi^s_n) \\
&\overset{(*)}{=} \sup\{\mathrm{Prob}(\cup_{i\in I} \Pi^s_i) \mid I \subseteq \mathbb{N}, I \text{ finite } \} \\
&= \sup\{\mathrm{Prob}(s, W_I, M) \mid W_I = \sqcup_{i\in I} B_i, I \text{ finite } \} \\
&= \sup\{\mathrm{Prob}(t, W_I, M) \mid W_I = \sqcup_{i\in I} B_i, I \text{ finite } \} \\
&= \mathrm{Prob}(t, W, M),
\end{aligned}$$

where the equality $(*)$ holds because of the following elementary property from measure theory: Let μ be a measure on some set, and let $A = \cup_{n\in\mathbb{N}} A_n$ be a measurable set which is a countable union of measurable sets. Then $\mu(A) = \sup\{\mu(\cup_{i\in I} A_i) \mid I \subseteq \mathbb{N}, I \text{ finite}\}$.

If $W = \sqcup_{i\in I} B_i$ contains arbitrary many blocks then there exists a countable index set $I_s \subseteq I$ and a saturated set $W_s = \sqcup_{i\in I_s} B_i$ such that $\mathrm{Prob}(s, W, M) = \mathrm{Prob}(s, W_s, M)$ using Lemma 5.3.2. For the same reason, there exists a countable index set $I_t \subseteq I$ and a corresponding saturated set $W_t = \sqcup_{i\in I_t} B_i$ with $\mathrm{Prob}(t, W, M) = \mathrm{Prob}(t, W_t, M)$. Hence $\mathrm{Prob}(s, W, M) = \mathrm{Prob}(s, W_s \cup W_t, M) = \mathrm{Prob}(t, W_s \cup W_t, M) = \mathrm{Prob}(t, W, M)$ since $W_s \cup W_t$ is countable, and that case we have already proved. □

Proof [of Theorem 5.3.30] The sufficiency part of the theorem holds trivially, having in mind Definition 5.3.12 and Remark 5.3.29, equation (5.29), since τ^* as well as $\tau^* a \tau^*$, for any $a \in A \setminus \{\tau\}$ is a saturated block and also each R-equivalence class is an R saturated set. Hence $\approx_{\{\tau\}}$ is at least as strong as $\approx_g$, $\approx_{\{\tau\}} \subseteq \approx_g$. For the necessity part, assume $s \approx_g t$ in a system $\langle S, A, \alpha\rangle$. Let R be a weak bisimulation according to Definition 5.3.12 such that $\langle s, t\rangle \in R$. By Proposition 5.3.14, we can assume that R is complete. By Lemma 5.3.35, we get that the transfer condition (5.29) of Remark 5.3.29 holds, and hence R is a weak bisimulation witnessing that $s \approx_{\{\tau\}} t$. □

5.4 Concluding remarks

In this chapter we have proposed a coalgebraic definition of weak bisimulation for action-type systems. For its justification we have considered the case of the familiar labelled transition systems and of generative probabilistic systems, and we have argued that the coalgebraic notion coincides with the concrete definitions.

Note however, that our approach does not provide a final solution to the weak bisimulation problem for coalgebras. Still, it provides a fresh view, and perhaps a way towards a solution. The main problem is namely that one has to come up with a suitable definition of a $*$-translation, in order to obtain a weak

bisimulation for a class of coalgebras of a given type. It would be better if a coalgebraic construction would automatically lead to $*$-translations. Obtaining $*$-translations is a topic for future research. The definition of composition from Chapter 6 is partially connected to this question.

Furthermore, most of the work and technical difficulties of this chapter were related to the correspondence result for generative probabilistic systems. We hope to have provided clear and complete proofs, and we have shown that the results of Baier and Hermanns extend to arbitrary infinite systems as well.

6

Simulation, coloring, composition, and paths for coalgebras

Semantic relations other than bisimulation and weak bisimulation are also often used for verification purposes. Such are for example simulations and trace equivalence. In this chapter we point to related work and briefly explain the state of the art of these semantic relations for probabilistic systems and coalgebras. The chapter collects some ideas and observations, and addresses further research.

In this chapter we collect some notes on various semantical issues. We take a general coalgebraic point of view and consider probabilistic systems as leading examples.

The chapter is divided into several sections. First, we focus on simulation relations in section 6.1. We briefly elaborate on existing work on coalgebraic simulations [JH03], as well as existing work on simulations of probabilistic systems [Seg95b, Bai98]. We show that the concrete definitions for probabilistic systems can be obtained from the coalgebraic one and we point out some open questions.

Next, we briefly mention colored trace semantics [GW96]. It is shown that colored trace semantics coincides with bisimilarity [GW96, AW05] for labelled transition systems and for the alternating probabilistic systems. Actually, in these two cases colored trace semantics is the same as colored transition semantics. We point out that colored transition semantics for coalgebras is actually behavior equivalence. We focus on this topic in Section 6.2.

Our third topic of interest is trace semantics. For some probabilistic systems there is a notion of trace semantics i.e. trace distribution [Seg95a, Seg95b, Bai98]. For coalgebras in general it is difficult to define what traces are. Interesting solutions for some classes of coalgebras have been recently proposed [Jac04, HJ05b, Jac05, HJ05a]. We believe that an important building block for defining trace semantics as well as weak bisimulation for coalgebras is the notion of a composite step. In a coalgebra, for each state, the transition func-

tion represents the one-step behavior of the state. In order to define traces or weak steps, one needs to consider sequences of steps i.e. composite steps. In Section 6.3 we discuss how composition of coalgebras, i.e. composition of steps in a coalgebra can be defined for some types of coalgebras.

We next emphasize the importance of the notion of paths. Consider labelled transition systems. A (finite) path π is an alternating sequence of states and labels of the form $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_n} s_n$. We notice that many of the semantic relations can be defined via a transformation on paths in the following way. Assume T is a function defined on paths. Then a T-semantics can be defined as: two states s and t are T-equivalent if and only if the image under T of the set of paths that start in s equals the image under T of the set of paths that start in t. Indeed, if the transformation T is the consistent coloring, then we get bisimilarity. If T is weak consistent coloring, then we get branching bisimilarity. If T maps a path π as above to its trace $a_1 \dots a_n$, then the semantic relation obtained is the trace equivalence. For probabilistic systems the notion of a path is also used in many occasions, for example we used it in Chapter 5. For these reasons, we investigate what a proper notion of a path in coalgebra might be. We focus on this in Section 6.4 and Section 6.5.

6.1 Simulation

Simulation can be viewed as uni-directional bisimulation in a sense that a state s can be simulated by a state t if every transition of s can be mimicked by a matching transition from t. The original definition for LTS is due to Milner [Mil80, Mil89].

Definition 6.1.1. *A relation $R \subseteq S \times T$ is a simulation between the LTSs $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ if and only if the following transfer condition holds whenever $\langle s, t \rangle \in R$:*

$$\text{if } s \xrightarrow{a} s', \text{ then there exists } t' \text{ with } t \xrightarrow{a} t' \text{ and } \langle s', t' \rangle \in R.$$

Hence, one of the two symmetric transfer conditions in the definition of a bisimulation (Definition 2.1.14) for LTS is dropped.

The definition of a simulation for simple Segala probabilistic automata, coalgebras of the functor $\mathcal{P}(A \times \mathcal{D})$ [SL94, Seg95b] follows the same lines.

Definition 6.1.2. *A relation $R \subseteq S \times T$ is a simulation between the two simple Segala systems $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ if and only if $\langle s, t \rangle \in R$ implies that*

$$\text{if } s \xrightarrow{a} \leadsto \mu, \text{ then there exists a distribution } \mu' \text{ with } t \xrightarrow{a} \leadsto \mu' \text{ and } \mu \equiv_R \mu'. \tag{6.1}$$

Please note, a bisimulation equivalence for simple Segala systems (Definition 2.2.3) was defined with the same transfer condition (6.1). This does not

mean that a simulation is a bisimulation which is not an equivalence. For bisimulations that are not equivalences between simple Segala systems one gets (as for the LTS, Definition 2.1.14) two symmetric transfer conditions, namely:

if $s \xrightarrow{a}\!\!\rightsquigarrow \mu$, *then there exists a distribution* μ' *with* $t \xrightarrow{a}\!\!\rightsquigarrow \mu'$ *and* $\mu \equiv_R \mu'$,

and

if $t \xrightarrow{a}\!\!\rightsquigarrow \mu$, *then there exists a distribution* μ' *with* $s \xrightarrow{a}\!\!\rightsquigarrow \mu'$ *and* $\mu \equiv_R \mu'$.

A notion of simulation has also been defined for the generative probabilistic systems (functor $\mathcal{D}(A \times \mathcal{I}d) + 1$) [Bai98].

Definition 6.1.3. *A relation* $R \subseteq S \times T$ *is a simulation between the two generative systems* $\langle S, \alpha\rangle$ *and* $\langle T, \beta\rangle$ *if and only if* $\langle s, t\rangle \in R$ *implies that either* $\alpha(s) = *$ *or*

if $s \rightsquigarrow \mu$, *then there exists a distribution* μ' *with* $t \rightsquigarrow \mu'$ *and* $\mu \equiv_{R,A} \mu'$.

Recall that $\equiv_{R,A} = \mathrm{Rel}(\mathcal{D}(A \times \mathcal{I}d))(R)$.

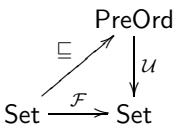
Definition 6.1.3 differs from the definition of bisimulation between generative systems (that need not be an equivalence) only in the fact that a terminating state s with $\alpha(s) = *$ can be simulated by any other state. Besides this special treatment of terminating states, a simulation according to Definition 6.1.3 is simply a bisimulation according to Definition 2.2.4. This way of defining simulations for generative probabilistic systems seems to be too restrictive.

Let us look at what exists on the topic of simulation relations for coalgebras. Following the definition of simulation for coalgebras by Jacobs and Hughes [JH03], we shall instantiate definitions for probabilistic systems. All the definitions and properties in this section are taken from [JH03].

The definition of simulation for coalgebras is based on a definition of *order of a functor* and of *lax relation lifting*, a relaxed form of relation lifting with respect to order of a functor. Therefore, the definition of simulation is parameterized by the order of the functor.

We denote by PreOrd the category of preorders $\langle X, \leq\rangle$ (with $\leq$ a reflexive and transitive relation on the set X) and order-preserving (monotone) functions between them. Let $\mathcal{F}$ be a set endofunctor. In order to define simulations for coalgebras of type $\mathcal{F}$ we need to have an order on the functor $\mathcal{F}$.

Definition 6.1.4. *Let* $\mathcal{F} : \mathsf{Set} \to \mathsf{Set}$. *An order on* $\mathcal{F}$ *is a functor* $\sqsubseteq\colon \mathsf{Set} \to \mathsf{PreOrd}$ *making the following diagram commute.*



where $\mathcal{U}$ denotes the forgetful functor from PreOrd *to* Set*, mapping a preorder to its underlying set.*

Hence, an order of a functor $\sqsubseteq$ is a collection of preorders $\sqsubseteq_X \subseteq \mathcal{F}X \times \mathcal{F}X$ indexed by sets, with the property that for any function $f : X \to Y$, $\mathcal{F}f : \mathcal{F}X \to \mathcal{F}Y$ is order preserving.

Given a functor $\mathcal{F}$ with an order $\sqsubseteq$, the notion of relation lifting $\mathrm{Rel}(\mathcal{F})$ (see Section 3.6.1) relaxes to *lax relation lifting* $\mathrm{Rel}_{\sqsubseteq}(\mathcal{F})$, by putting, for a relation R,

$$\begin{aligned}\mathrm{Rel}_{\sqsubseteq}(\mathcal{F})(R) &= \sqsubseteq \circ \mathrm{Rel}(\mathcal{F})(R) \circ \sqsubseteq && (6.2)\\ &= \{\langle u, v\rangle \mid \exists u', v' : u \sqsubseteq u', \langle u', v'\rangle \in \mathrm{Rel}(\mathcal{F})(R), v' \sqsubseteq v\} && (6.3)\\ &= \{\langle u, v\rangle \mid \exists r \in \mathcal{F}R : u \sqsubseteq \mathcal{F}(\pi_1)(r), \mathcal{F}(\pi_2)(r) \sqsubseteq v\}. && (6.4)\end{aligned}$$

Just like bisimulation can be defined via relation liftings (see Lemma 3.6.4), a simulation is defined via the lax relation liftings.

Definition 6.1.5. *Let $\mathcal{F}$ be a* Set *endofunctor, with an order $\sqsubseteq$. A relation $R \subseteq S \times T$ is a simulation between the $\mathcal{F}$ coalgebras $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if*

$$\langle s, t\rangle \in R \implies \langle \alpha(s), \beta(t)\rangle \in \mathrm{Rel}_{\sqsubseteq}(\mathcal{F})(R). \tag{6.5}$$

This simple change from the definition of bisimulation to the definition of simulation suffices to obtain the expected properties of simulations and similarity for any coalgebra. In order to instantiate Definition 6.1.5 to concrete systems, one needs to provide an order on the functor. For example, for the powerset functor $\mathcal{P}$, it is not hard to see that the inclusion order $\subseteq_S$ on $\mathcal{P}S$ provides an order $\subseteq$ on $\mathcal{P}$. In general, the following holds.

Lemma 6.1.6. *If $\mathcal{F}$ has an order $\sqsubseteq$ and $\mathcal{G}$ is any functor, than $\mathcal{F}\mathcal{G}$ inherits an order $\sqsubseteq'$ from $\mathcal{F}$ defined by $\sqsubseteq'_S = \sqsubseteq_{\mathcal{G}S}$.* □

We shall denote the inherited order the same as the original order. When Definition 6.1.5 is instantiated to LTS, functor $\mathcal{P}(A \times \mathcal{I}d)$ one uses the inclusion order $\subseteq$ and obtains the usual notion of simulation (Definition 6.1.1). Namely, a relation $R \subseteq S \times T$ is a simulation between the LTS $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if $\langle s, t\rangle \in R$ implies that there exist $X \subseteq A \times S$ and $Y \subseteq A \times T$ such that

$$\alpha(s) \subseteq X,\ \langle X, Y\rangle \in \mathrm{Rel}(\mathcal{P}(A \times \mathcal{I}d))(R),\ Y \subseteq \beta(t)$$

which by the the definition and properties of relation lifting (see Section 3.6.1 and Section 3.6.2) can easily be seen to be equivalent to the condition from Definition 6.1.1.

In [JH03] besides the LTS many other examples are discussed, but among them there are no probabilistic systems.

We will instantiate this coalgebraic definition of simulation from [JH03] to some probabilistic systems, in fact to the simple Segala systems and the generative probabilistic systems mentioned above.

For the simple Segala systems, functor $\mathcal{P}(A\times\mathcal{D})$, we again use the inclusion order $\subseteq$ just like for LTS. Therefore, it is no surprise that the transfer condition (6.5) instantiated to $\mathcal{P}(A \times \mathcal{D})$ with an order $\subseteq$, amounts to the transfer condition from Definition 6.1.2.

In order to derive a notion of simulation for the generative probabilistic systems, functor $\mathcal{D}(A \times S) + 1$, we first need a definition of an order on $\mathcal{D}$, or rather on $\mathcal{D} + 1$. Actually, any order $\sqsubseteq$ on $\mathcal{D}$ can be extended to an order on $\mathcal{D} + 1$ in a trivial way by adding $\langle *, *\rangle$ to the order. One order on $\mathcal{D}$ can be obtained by the following lemma.

Lemma 6.1.7. *If there is an order $\sqsubseteq$ on a functor $\mathcal{F}$, and a functor $\mathcal{G}$ is such that there is a natural transformation $\tau : \mathcal{G} \Longrightarrow \mathcal{F}$, then $\sqsubseteq$ induces an order on $\mathcal{G}$, denoted also by $\sqsubseteq$. For $u, v \in \mathcal{G}X$ we have*

$$u \sqsubseteq v \iff \tau_X(u) \sqsubseteq \tau_X(v).$$

□

There is a natural transformation $\mathrm{supp} : \mathcal{D} \Longrightarrow \mathcal{P}$ mapping any distribution to its support set, i.e., $\mathrm{supp}_X(\mu) = \mathrm{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$ for any $\mu \in \mathcal{D}X$. Hence, from Lemma 6.1.7, we get an order $\subseteq$ on $\mathcal{D}$, induced by the inclusion order on $\mathcal{P}$. For $\mu, \nu \in \mathcal{D}_X$, it is defined by

$$\mu \subseteq \nu \iff \mathrm{supp}(\mu) \subseteq \mathrm{supp}(\nu).$$

We can extend this order to an order on $\mathcal{D} + 1$ as mentioned before.

From condition (6.5) and equation (6.3), we get that a relation $R \subseteq S \times T$ is a $\subseteq$-simulation between the generative probabilistic systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if $\langle s, t\rangle \in R$ implies that, either both $\alpha(s) = *$ and $\beta(t) = *$, or $\alpha(s) = \mu$, $\beta(t) = \nu$ and

$$\exists \mu', \nu' : \mathrm{supp}(\mu) \subseteq \mathrm{supp}(\mu'),\ \mu' \equiv_{R,A} \nu',\ \mathrm{supp}(\nu') \subseteq \mathrm{supp}(\nu). \tag{6.6}$$

Without deriving equivalent conditions for $\subseteq$-simulations, we note that the notion of $\subseteq$-simulation is rather general, perhaps too general. Note that given two coalgebras on the same set of states S, any reflexive relation R on S that satisfies

$$\langle s, t\rangle \in R \implies (\alpha(s) = \beta(t) = *) \vee (\alpha(s) = \mu \wedge \beta(t) = \nu \wedge \mathrm{supp}(\mu) \subseteq \mathrm{supp}(\nu))$$

is an $\subseteq$-simulation. Indeed, if $\langle s, t\rangle \in R$, $\alpha(s) = \mu$, $\beta(t) = \nu$, and $\mathrm{supp}(\mu) \subseteq \mathrm{supp}(\nu)$, then by taking $\mu' = \nu' = \mu$ we have satisfied condition (6.6).

The simulations from Definition 6.1.3 can also be obtained from the coalgebraic definition. Consider, for any set X, the order $\sqsubseteq_X$ on $(\mathcal{D} + 1)X \cong \mathcal{D}X + 1$, defined by

$$x \sqsubseteq_X y \iff x = y \in \mathcal{D}X \vee x = *.$$

Hence, $\sqsubseteq_X$ is simply the equality on distributions and $*$ is added as the smallest element. The collection $\sqsubseteq$ of all $\sqsubseteq_X$ forms an order of the functor $\mathcal{D}+1$. The corresponding $\sqsubseteq$-simulation relations satisfy exactly the same transfer condition as the one from Definition 6.1.3.

Another definition of simulation for generative probabilistic systems can be obtained if we change the distribution functor $\mathcal{D}$ to the sub-probability distribution functor $\mathcal{D}_{\leq}$. This functor is defined following the same lines as for $\mathcal{D}$ only there is no requirement that the sum of the probabilities equals 1. Instead, the sum should be less than or equal to 1. This functor has attracted some attention lately (e.g. [HJ05a]). The generative probabilistic systems, including the possibility of termination, can be modelled as coalgebras of type $\mathcal{D}_{\leq}(A \times \mathcal{I}d)$. A non-trivial point-wise order $\leq$ exists on $\mathcal{D}_{\leq}$, defined for $\mu, \nu \in \mathcal{D}_{\leq}X$ by

$$\mu \leq \nu \iff \forall x \in X : \mu(x) \leq \nu(x)$$

where $\leq$ on the right-hand side denotes the ordering of the real numbers. Using this order, from (6.5) and (6.4) a relation $R \subseteq S \times T$ is an $\leq$-simulation between the generative systems $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ if and only if $\langle s, t \rangle \in R$ and $\alpha(s) = \mu$, $\beta(t) = \nu$ implies that

$$\exists \rho \in \mathcal{D}_{\leq}R : \mathcal{D}_{\leq}\pi_1(\rho) \geq \mu \wedge \mathcal{D}_{\leq}\pi_2(\rho) \leq \nu$$

which is equivalent to

$$\exists \rho \in \mathcal{D}_{\leq}R : \sum_y \rho(x, y) \geq \mu(x) \wedge \sum_x \rho(x, y) \leq \nu(y).$$

Hence, different possibilities for orders on a functor exist. They lead to different notions of simulation relations. It might be interesting to study whether some orders are in any sense better than others. A related question is the one of existence of a canonical order on a given functor. For example, for the power set functor $\mathcal{P}$, the inclusion order seems to be canonical.

Investigating weak simulations in the context of Chapter 5 and [JH03] is also an interesting open question.

6.2 Colored transitions

A coincidence result for bisimilarity and so-called colored trace equivalence for LTS was given by van Glabbeek and Weijland [GW96]. Andova and Willemse [AW05] extended the notion of colored trace equivalence to one type of probabilistic systems, the alternating systems of Hansson [Han91, HJ94]. The same authors showed that bisimilarity for the alternating systems coincides with the colored trace equivalence.

In an LTS, two states are colored trace equivalent if and only if they are colored the same by some consistent coloring of the states of the system. A coloring

of the states of a system is called consistent if two states are colored the same only when their colored traces are the same. Colored trace equivalence coincides with bisimilarity [GW96].

It is easy to see, in the case of LTS and in the case of the alternating probabilistic systems, that a coloring is consistent if and only if two states are colored the same only when their transitions are colored the same i.e. they have the same colored traces of length 1.

Here we demonstrate that bisimilarity on any coalgebra of type $\mathcal{F}$, for $\mathcal{F}$ preserving weak pullbacks, is characterized via colored transitions. The characterization is a trivial reformulation of already known notions and facts which we present in the sequel. We stress that we straightforwardly generalize the notions of colored transition and consistent coloring (in terms of colored transition) from [GW96] to coalgebras.

Let $\langle S, \alpha\rangle$ be an $\mathcal{F}$ coalgebra. Let C be a set of colors. A *coloring* of the system $\langle S, \alpha\rangle$ is any function $\mathrm{col} : S \to C$. Note that, by applying the functor $\mathcal{F}$, any coloring (of states) function $\mathrm{col} : S \to C$ extends to a coloring of transitions $\mathcal{F}\,\mathrm{col} : \mathcal{F}S \to \mathcal{F}C$.

An element $\langle c, \alpha_c\rangle \in C \times \mathcal{F}C$ is a *colored transition* of $\langle S, \alpha\rangle$ if and only if there exists a state $s \in S$ such that

$$(\mathrm{col} \times \mathcal{F}\,\mathrm{col})(\langle s, \alpha(s)\rangle) = \langle \mathrm{col}(s), \mathcal{F}\,\mathrm{col}(\alpha(s))\rangle = \langle c, \alpha_c\rangle.$$

Definition 6.2.1. *A coloring* col *of the system* $\langle S, \alpha\rangle$ *is consistent if and only if two states are colored the same only when their transitions are colored the same, i.e.,*

$$\mathrm{col}(s) = \mathrm{col}(t) \implies \mathcal{F}\,\mathrm{col}(\alpha(s)) = \mathcal{F}\,\mathrm{col}(\alpha(t)). \tag{6.7}$$

It turns out that consistent coloring is nothing else but a coalgebraic homomorphism.

Lemma 6.2.2. *A function* $\mathrm{col} : S \to C$ *is a consistent coloring of the system of type* $\mathcal{F}$, $\langle S, \alpha\rangle$ *if and only if a system* $\langle C, \alpha_C\rangle$ *of type* $\mathcal{F}$ *can be defined such that* col *is a coalgebra homomorphism from* $\langle S, \alpha\rangle$ *to* $\langle C, \alpha_C\rangle$*, as shown in the next diagram:*

$$\begin{array}{ccc} S & \xrightarrow{\mathrm{col}} & C \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\exists\alpha_C} \\ \mathcal{F}S & \xrightarrow{\mathcal{F}\,\mathrm{col}} & \mathcal{F}C \end{array}$$

Proof Assume α_C exists such that $\mathrm{col} : S \to C$ is a homomorphism from $\langle S, \alpha\rangle$ to $\langle C, \alpha_C\rangle$. Then

$$\begin{aligned} \mathrm{col}(s) = \mathrm{col}(t) &\implies \alpha_C(\mathrm{col}(s)) = \alpha_C(\mathrm{col}(t)) \\ &\implies \mathcal{F}\,\mathrm{col}(\alpha(s)) = \mathcal{F}\,\mathrm{col}(\alpha(t)). \end{aligned}$$

where the last implication holds by the homomorphism assumption. Hence, col is consistent.

For the opposite, assume $\mathrm{col} : S \to C$ is a consistent coloring of the $\mathcal{F}$ coalgebra $\langle S, \alpha \rangle$. We can assume that the coloring is surjective, since the color set C can always be restricted to a smaller set of used colors without affecting the consistency condition. Let $c \in C$. Put

$$\alpha_C(c) = \mathcal{F}\,\mathrm{col}(\alpha(s))$$

for some $s \in \mathrm{col}^{-1}(c)$. We need to check that α_C is well defined. Let $s, t \in \mathrm{col}^{-1}(c)$. Then $\mathrm{col}(s) = \mathrm{col}(t)$ and by the consistency condition $\mathcal{F}\,\mathrm{col}(\alpha(s)) = \mathcal{F}\,\mathrm{col}(\alpha(t))$, i.e. α_C is determined independently of the choice of representative in $\mathrm{col}^{-1}(c)$. Furthermore, col is a homomorphism by the definition of α_C, i.e.

$$\alpha_C \circ \mathrm{col} = \mathcal{F}\,\mathrm{col} \circ \alpha$$

which completes the proof. □

Next we define a relation *colored transition equivalence*, on the states of a system of type $\mathcal{F}$.

Definition 6.2.3. *Let $\langle S, \alpha \rangle$ be an $\mathcal{F}$ coalgebra. The states s and t in S are colored transition equivalent, notation $s \equiv_C t$, if and only if s and t are colored with the same color by some consistent coloring of $\langle S, \alpha \rangle$.*

The next lemma shows that bisimilarity coincides with colored transition equivalence, for coalgebras of weak pullback preserving functors.

Lemma 6.2.4. *Let $\langle S, \alpha \rangle$ be an $\mathcal{F}$ coalgebra and let $s, t \in S$ be two states. If $\mathcal{F}$ preserves weak pullbacks, then*

$$s \sim t \iff s \equiv_C t.$$

Proof Assume $s \sim t$. Since $\mathcal{F}$ preserves weak pullbacks, there exists an equivalence bisimulation R such that $\langle s, t \rangle \in R$. Consider the coloring $c : S \to S/R$ "coloring" each state with its R-equivalence class. According to [Rut00, Proposition 5.8] there exists a transition structure $\alpha_{S/R}$ on S/R making c a homomorphism from $\langle S, \alpha \rangle$ to $\langle S/R, \alpha_{S/R} \rangle$. Now by Lemma 6.2.2, c is consistent and furthermore $c(s) = c(t)$. Therefore $s \equiv_C t$.

For the opposite, assume $s \equiv_C t$. Let col be a consistent coloring of $\langle S, \alpha \rangle$ such that $\mathrm{col}(s) = \mathrm{col}(t)$. By Lemma 6.2.2, we have that there exists a transition structure α_C on C such that $\mathrm{col} : S \to C$ is a homomorphism. Consider the (equivalence) relation R on S defined as the kernel of col, i.e.,

$$R = \ker(\mathrm{col}) = \{\langle s, t \rangle \mid \mathrm{col}(s) = \mathrm{col}(t)\}.$$

Since the functor $\mathcal{F}$ preserves weak pullbacks and R is a kernel of a homomorphism, we get that R is a bisimulation (see [Rut00, Proposition 5.7]). From $\langle s, t \rangle \in R$ we have $s \sim t$. □

It is obvious that $\equiv_C$ is reflexive (the identity coloring is consistent) and symmetric, simply by definition. By Lemma 6.2.4, for coalgebras of weak pullback preserving functors, the name colored transition equivalence is justified, i.e. $\equiv_C$ is an equivalence indeed, since bisimilarity is an equivalence. However, the colored transition equivalence is an equivalence even if the bisimilarity is not, since it coincides with behavior equivalence. Recall that we denoted behavioral equivalence by $\approx$ (see Chapter 4, Definition 4.3.1).

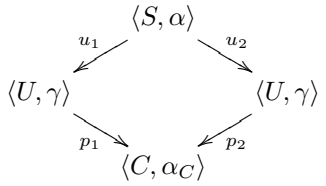
Lemma 6.2.5. *Let $\langle S, \alpha\rangle$ be an $\mathcal{F}$ coalgebra and let $s, t \in S$ be two states. Then*

$$s \approx t \iff s \equiv_C t.$$

Before we give the proof, we note that the same property in a different formulation is a known characterization of behavior equivalence on a coalgebra (see, e.g., [Pat03, Proposition 2.3.3]).

Proof (of Lemma 6.2.5) It is clear that colored transition equivalence implies behavior equivalence, since if $s \equiv_C t$ via a consistent coloring col, then $\langle C, \mathrm{col}, \mathrm{col}\rangle$ is a cocongruence on $\langle S, \alpha\rangle$ that identifies s and t.

Assume $s \approx t$, and further assume that the cocongruence $\langle U, u_1, u_2\rangle$ identifies them, i.e. $u_1(s) = u_2(t)$. Since $\langle U, u_1, u_2\rangle$ is a cocongruence, $\langle\langle U, \gamma\rangle, u_1, u_2\rangle$ is a cospan in $\mathsf{Coalg}_{\mathcal{F}}$. Moreover, since u_1 and u_2 have the same domain, $\langle\langle S, \alpha\rangle, u_1, u_2\rangle$ is a span in $\mathsf{Coalg}_{\mathcal{F}}$. Take the pushout $\langle\langle C, \alpha_C\rangle, p_1, p_2\rangle$ of $\langle\langle S, \alpha\rangle, u_1, u_2\rangle$. We have the commuting diagram



The pushout exists since all colimits exist in $\mathsf{Coalg}_{\mathcal{F}}$. Moreover it is obtained from the pushout $\langle C, p_1, p_2\rangle$ of $\langle S, u_1, u_2\rangle$ in Set and equipped with the unique corresponding transition structure α_C. From the construction of the pushout in Set we have that $C = U/\theta$ for θ being the smallest equivalence on U generated by the pairs $\{\langle u_1(s), u_2(s)\rangle \mid s \in S\}$, and p_1, p_2 are in this case a single map, the canonical projection mapping each element to its θ-class.

Since $u_1(s) = u_2(t)$ we get that both $\langle u_2(t), u_2(s)\rangle \in \theta$ and $\langle u_1(t), u_1(s)\rangle \in \theta$. Hence $p_1 \circ u_1(s) = p_1 \circ u_1(t)$, i.e. for the map $\mathrm{col} = p_1 \circ u_1 = p_2 \circ u_2$ we have $\mathrm{col}(s) = \mathrm{col}(t)$. Moreover, col is a homomorphism from $\langle S, \alpha\rangle$ to $\langle C, \alpha_C\rangle$ and hence a consistent coloring which gives us $s \equiv_C t$. □

We obtained that two states are behavior equivalent if and only if there exists a consistent coloring that colors them the same. So, the colored transition equivalence is just another way of defining behavior equivalence and the presented

material of this section provided another way to prove that behavior equivalence and bisimilarity coincide for weak pullback preserving functors.

Furthermore, studying the connection between consistency of colorings and relation liftings might bring some more insight. Consider the condition (6.7), i.e., the consistency condition of a coloring. The condition can be restated into

$$\langle s,t\rangle \in ker(col) \implies \langle \alpha(s),\alpha(t)\rangle \in ker(\mathcal{F}\,\mathrm{col}). \tag{6.8}$$

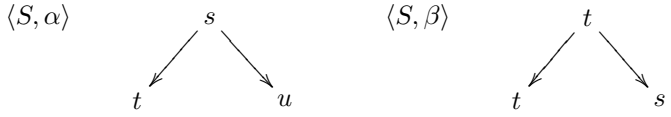
It is also worth stating that the behavior equivalence $\approx$ on an $\mathcal{F}$-coalgebra satisfies

$$\approx \;=\; \bigcup_{\mathrm{col}:\langle S,\alpha\rangle\to\langle C,\alpha_C\rangle} \ker(\mathrm{col}) \tag{6.9}$$

for any homomorphism col to any other $\mathcal{F}$ coalgebra. Then the condition (6.8) is something like a transfer condition for cocongruences, if we consider them as relations, i.e. as sets of pairs of states that they identify.

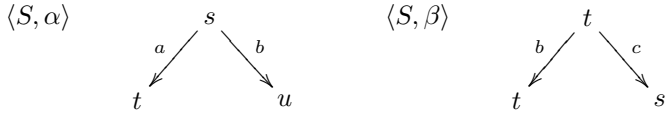
6.3 Composition of coalgebras

Consider a transition system $\langle S,\alpha : S\to\mathcal{P}S\rangle$ with $s,t,u\in S$. The outgoing transitions from the state s are shown in the left diagram below.

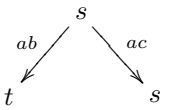


Moreover, assume that there is a TS $\langle S,\beta : S\to\mathcal{P}S\rangle$ ($\beta=\alpha$ is possible) in which the state t allows the transitions shown in the right diagram above, and the state u is terminating. Transitions that correspond to the sequential composition of $\langle S,\alpha\rangle$ and $\langle S,\beta\rangle$ from the state s are as shown below.

In this case, we compose by composing the transition relation $\longrightarrow$. Similarly, in the case of LTS, we consider $\langle S,\alpha\rangle$ and $\langle S,\beta\rangle$ with the transitions from s and t in the two systems as below, and u a terminating state.



The composition in state s is then described by the following transitions (now with labels from A^2).

$\langle S, \alpha \cdot \beta \rangle$ 

In general, if two systems $\langle S, \alpha : S \to \mathcal{F}S \rangle$ and $\langle S, \beta : S \to \mathcal{F}S \rangle$ are given, then we wonder which system (on S) behaves as if a step from $\langle S, \alpha \rangle$ is followed by a step from $\langle S, \beta \rangle$. We wish to define such a system $\langle S, \alpha \cdot \beta \rangle$. We could always define this composition to be of type $\mathcal{F}\mathcal{F}$, by $\alpha \cdot \beta = \mathcal{F}\beta \circ \alpha$, i.e.

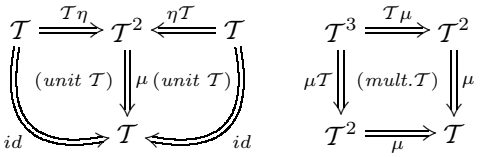
$$S \xrightarrow{\alpha} \mathcal{F}S \xrightarrow{\mathcal{F}_\beta} \mathcal{F}\mathcal{F}S. \tag{6.10}$$

However, in the case of transition systems we get a composed system of type $\mathcal{P}$ and not $\mathcal{P}\mathcal{P}$, and in the case of LTS we get a composed system of type $\mathcal{P}(A^2 \times \mathcal{I}d)$ and not of type $\mathcal{P}(A \times \mathcal{I}d)\mathcal{P}(A \times \mathcal{I}d)$. This is due to the richer structure of $\mathcal{P}$, namely it is a *monad*. Moreover there is a *distributive law* $\pi : (A \times \mathcal{I}d)\mathcal{P} \Longrightarrow \mathcal{P}(A \times \mathcal{I}d)$. Distributive laws have various applications in the theory of coalgebras (c.f. [Bar04]). In this section we shall see how composition of systems can be defined for systems of type $\mathcal{T}\mathcal{F}$ where $\mathcal{T}$ is a monad, with a distributive law $\lambda : \mathcal{F}\mathcal{T} \Longrightarrow \mathcal{T}\mathcal{F}$.

6.3.1 Monads and distributive laws

We start by introducing the notions that we need for the sequel.

Definition 6.3.1. *A monad in a category* $\mathbf{C}$ *is a triple* $\langle T, \eta, \mu \rangle$ *where* $\mathcal{T}$ *is a* $\mathbf{C}$ *endofunctor, and* $\eta : \mathcal{I}d \Longrightarrow \mathcal{T}$, $\mu : \mathcal{T} \circ \mathcal{T} \Longrightarrow \mathcal{T}$ *are natural transformations, called the unit and the multiplication, respectively, such that the following diagrams commute.*



The two parts of the left diagram are the unit laws, and the right diagram is the multiplication law, or the associativity, of the monad.

Example 6.3.2. A typical example of a monad in Set is the powerset monad $\langle \mathcal{P}, \{\}, \bigcup \rangle$ where $\{\} : \mathcal{I}d \Longrightarrow \mathcal{P}$ is the singleton natural transformation given by $\{\}_X(x) = \{x\}$, and $\bigcup : \mathcal{P}^2 \Longrightarrow \mathcal{P}$ is the union natural transformation given by $\bigcup_X(Y) = \bigcup_{Z \in Y} Z$ for any $Y \in \mathcal{P}\mathcal{P}X$.

The distribution functor can also be equipped with a monad structure, namely $\langle \mathcal{D}, \eta, \mu \rangle$ is a monad for $\eta : \mathcal{I}d \Longrightarrow \mathcal{D}$ being the Dirac natural transformation

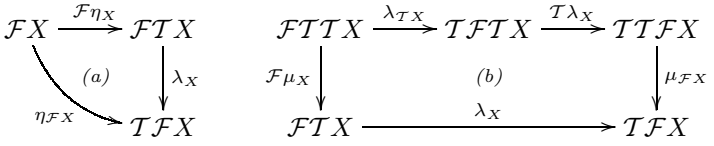
given by $\eta_X(x) = \mu^1_x$, and the multiplication $\mu : \mathcal{D}^2 \Longrightarrow \mathcal{D}$ is given by $\mu_X(\nu) = \bar{\nu}$ for $\nu \in \mathcal{D}\mathcal{D}X$ and $\bar{\nu} \in \mathcal{D}X$ defined by

$$\bar{\nu}(x) = \sum_{\xi \in \mathcal{D}X} \nu(\xi) \cdot \xi(x).$$

Simple derivations suffice to check that the unit and the multiplication laws hold in this case as well.

Let $\mathcal{F}$ and $\mathcal{G}$ be any endofunctors on a category $\mathbf{C}$. A distributive law of $\mathcal{F}$ over $\mathcal{G}$ is a natural transformation $\lambda : \mathcal{F}\mathcal{G} \Longrightarrow \mathcal{G}\mathcal{F}$. For the sequel we will use distributive laws of a functor over a monad, whose definition we give next.

Definition 6.3.3. *Let $\mathcal{F}$ be a functor and $\mathcal{T}$ a monad. A plain distributive law of $\mathcal{F}$ over $\mathcal{T}$ is a distributive law of $\mathcal{F}$ over the functor $\mathcal{T}$. A distributive law of $\mathcal{F}$ over the monad $\mathcal{T}$ is a natural transformation $\lambda : \mathcal{F}\mathcal{T} \Longrightarrow \mathcal{T}\mathcal{F}$ that preserves the monad structure, i.e., the following diagrams commute.*



Example 6.3.4. Let $\mathcal{F} = \underline{A} \times \mathcal{I}d$ and $\mathcal{T} = \mathcal{D}$ be the distribution monad. Then there exists a distributive law $\lambda : \mathcal{F}\mathcal{T} \Longrightarrow \mathcal{T}\mathcal{F}$ i.e. $\lambda : \underline{A} \times \mathcal{D} \Longrightarrow \mathcal{D}(\underline{A} \times \mathcal{I}d)$, given by

$$\lambda_X(\langle a, \mu\rangle)(\langle b, s\rangle) = \begin{cases} \mu(s) & a = b \\ 0 & \text{otherwise.} \end{cases}$$

This distributive law preserves the monad structure. Note that $\lambda_X(\langle a, \mu\rangle) = \mu^1_a \times \mu$, for $\times$ denoting the product of distributions, and μ^1_a the Dirac distribution for $a \in A$. We already used this natural transformation in Chapter 4 (page 109).

When dealing with the powerset monad, a distributive law comes for free, as in the next lemma.

Lemma 6.3.5. *([Jac04, HJ05b]) Let $\mathcal{F}$ be any weak pullback preserving functor. Then there exists a distributive law $\pi : \mathcal{F}\mathcal{P} \Longrightarrow \mathcal{P}\mathcal{F}$ of $\mathcal{F}$ over the powerset monad $\mathcal{P}$, called a power law. The power law π is given by*

$$\pi_X(v) = \{u \in \mathcal{F}X \mid \langle u, v\rangle \in \mathrm{Rel}(\mathcal{F})(\in_X)\} \tag{6.11}$$

for any set X and $v \in \mathcal{F}\mathcal{P}X$. □

Example 6.3.6. In particular, for $\mathcal{F} = \underline{A} \times \mathcal{I}d$, Lemma 6.3.5 provides us with a power law $\pi : \underline{A} \times \mathcal{P} \Longrightarrow \mathcal{P}(\underline{A} \times \mathcal{I}d)$ which according to the definition of relation lifting for $\underline{A} \times \mathcal{I}d$ is given by

$$\pi_X(\langle a, X'\rangle) = \{\langle a, x\rangle \mid x \in X'\} \tag{6.12}$$

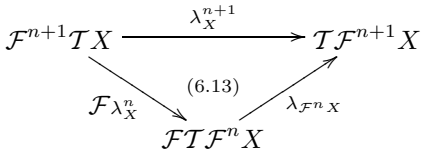
for any $a \in A$ and any $X' \subseteq X$.

Recently, Hasuo and Jacobs [HJ05a] have shown existence of a distributive law $\delta : \mathcal{FD} \Longrightarrow \mathcal{DF}$ for polynomial functors $\mathcal{F}$, providing a generalization of Example 6.3.4 in the sense of Lemma 6.3.5.

Assume $\mathcal{T}$ is a monad, $\mathcal{F}$ a functor, and assume there exists a (plain) distributive law $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$. Following [Jac04] we define families of maps $\lambda^n_X : \mathcal{F}^n\mathcal{T}X \to \mathcal{TF}^nX$, indexed by sets, for all $n \in \mathbb{N}$ by

$$\lambda^0_X = id_{\mathcal{T}X}, \quad \lambda^{n+1}_X = \lambda_{\mathcal{F}^nX} \circ \mathcal{F}\lambda^n_X \tag{6.13}$$

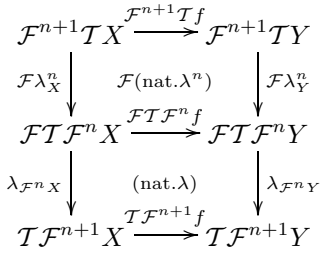
i.e.



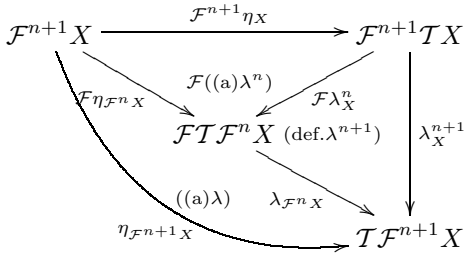
Lemma 6.3.7. *Let $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$ be a (plain) distributive law. For all $n \in \mathbb{N}$, from (6.13) we get a (plain) distributive law*

$$\lambda^n : \mathcal{F}^n\mathcal{T} \Longrightarrow \mathcal{TF}^n.$$

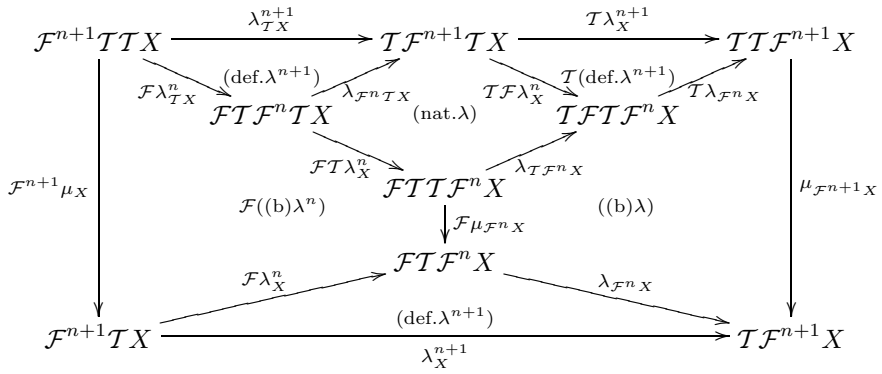
Proof We first show the naturality of λ^n for $n \in \mathbb{N}$, by induction. For $n = 0$ the statement is trivial saying that $id : \mathcal{T} \Longrightarrow \mathcal{T}$. For $n = 1$ the statement is the naturality of λ. Assume λ^n is natural, and let $f : X \to Y$. Then we have that the following diagram commutes



giving the naturality of λ^{n+1}. It remains to show that if λ preserves the monad structure of $\mathcal{T}$, then λ^n also does. We show this also by induction on n. Again, the case $n = 0$ is trivial, and the case $n = 1$ is the statement for λ. We need to show that both (a) and (b) from Definition 6.3.3 are satisfied for λ^{n+1} assuming that they are for λ^i if $i \leq n$. We obtain (a) from the following diagram



and (b) from the diagram below



which completes the proof. □

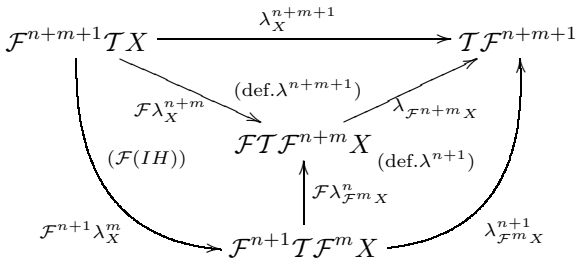
Lemma 6.3.8. *For λ^k defined by (6.13) and for all natural numbers $n, m \in \mathbb{N}$, it holds that*

$$\lambda^{n+m}_X = \lambda^n_{\mathcal{F}^m X} \circ \mathcal{F}^n \lambda^m_X. \tag{6.14}$$

Proof We prove the property by induction on n. We have

$$\lambda^{0+m}_X = \lambda^m_X = id \circ \lambda^m_X = \lambda^0_{\mathcal{F}^m X} \circ \mathcal{F}^0 \lambda^m_X.$$

We show that if it holds for the pair $\langle n, m\rangle$, then it does for $\langle n+1, m\rangle$, by the commutativity of the following diagram.



□

Example 6.3.9. Consider again the setting of Example 6.3.6, and the given power law $\pi : \mathcal{F}\mathcal{P} \Longrightarrow \mathcal{P}\mathcal{F}$ for $\mathcal{F} = \underline{A} \times \mathcal{I}d$. Since

$$(\underline{A} \times \mathcal{I}d)^n \cong \underline{A^n} \times \mathcal{I}d,$$

from Lemma 6.3.8, we get n-fold power law $\pi^n : \underline{A^n} \times \mathcal{P} \Longrightarrow \mathcal{P}(\underline{A^n} \times \mathcal{I}d)$, for each $n \in \mathbb{N}$. According to Equation (6.13), we can derive that it is given by

$$\pi^n_X(\langle w, X'\rangle) = \{\langle w, x\rangle \mid x \in X'\}$$

for $w \in A^n$ and $X' \subseteq X$.

6.3.2 Composition

We can now define composition of coalgebras of type $\mathcal{TF}$ for a monad $\mathcal{T}$ with a distributive law $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$.

Let S be a given set. We consider the set of all systems with carrier set S of type $\mathcal{TF}^n$, for some $n \in \mathbb{N}$. Let $\langle S, \alpha\rangle$ and $\langle S, \beta\rangle$ be two such systems, $\alpha : S \to \mathcal{TF}^k S,\ \beta : S \to \mathcal{TF}^m S$. We define

$$\langle S, \gamma\rangle = \langle S, \alpha\rangle \cdot \langle S, \beta\rangle$$

for $\gamma : S \to \mathcal{TF}^{k+m} S$ as given by the following diagram

$$\begin{array}{ccccccc} S & \xrightarrow{\alpha} & \mathcal{TF}^k S & \xrightarrow{\mathcal{TF}^k\beta} & \mathcal{TF}^k\mathcal{TF}^m S & \xrightarrow{\mathcal{T}\lambda^k_{\mathcal{F}^m S}} & \mathcal{T}^2\mathcal{F}^{k+m} S \\ & \searrow^{\gamma} & & & & & \downarrow{\scriptstyle \mu_{\mathcal{F}^{k+m}S}} \\ & & & & & & \mathcal{TF}^{k+m} S \end{array} \tag{6.15}$$

The system $\langle S, \gamma\rangle$ is called the composition of $\langle S, \alpha\rangle$ and $\langle S, \beta\rangle$. When the carrier set is clear from the context, we shall often just write $\gamma = \alpha \cdot \beta$. Note that if $\mathcal{T}$ is the identity monad, then one obtains the obvious definition of composition as in (6.10).

The next lemma shows that the composition is a monoid operation.

Lemma 6.3.10. *Let $\mathcal{T}$ be a monad, $\mathcal{F}$ a functor, and assume that there exists a distributive law $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$. The following hold.*

(*i*) *The composition of systems is associative.*

(*ii*) *The system $\langle S, \eta_S : S \to \mathcal{T}S\rangle$ is a unit for the composition of systems.*

Proof

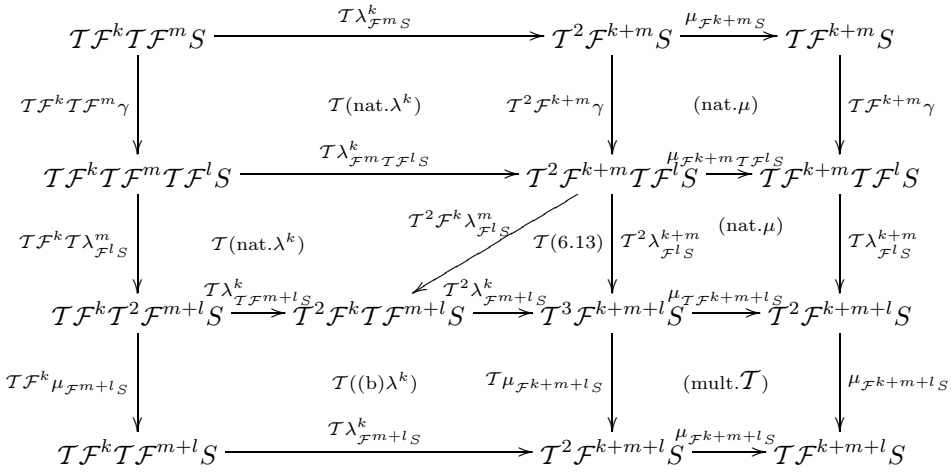
(*i*) Let $\langle S, \alpha\rangle, \langle S, \beta\rangle$ and $\langle S, \gamma\rangle$ be such that $\alpha : S \to \mathcal{TF}^k S,\ \beta : S \to \mathcal{TF}^m S$ and $\gamma : S \to \mathcal{TF}^l S$. By the definition of the composition we have

$$\begin{aligned} (\alpha \cdot \beta) \cdot \gamma &= (\mu_{\mathcal{F}^{k+m}S} \circ \mathcal{T}\lambda^k_{\mathcal{F}^m S} \circ \mathcal{TF}^k\beta \circ \alpha) \cdot \gamma \\ &= \mu_{\mathcal{F}^{k+m+l}S} \circ \mathcal{T}\lambda^{k+m}_{\mathcal{F}^l S} \circ \mathcal{TF}^{k+m}\gamma \circ \mu_{\mathcal{F}^{k+m}S} \circ \mathcal{T}\lambda^k_{\mathcal{F}^m S} \circ \mathcal{TF}^k\beta \circ \alpha \end{aligned}$$

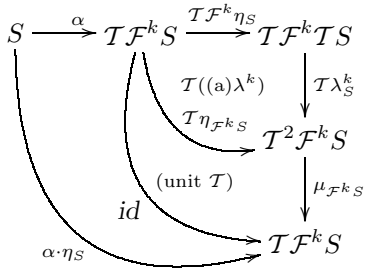
and

$$\begin{aligned} \alpha \cdot (\beta \cdot \gamma) &= \alpha \cdot (\mu_{\mathcal{F}^{m+l}S} \circ \mathcal{T}\lambda^m_{\mathcal{F}^l S} \circ \mathcal{TF}^m\gamma \circ \beta) \\ &= \mu_{\mathcal{F}^{k+m+l}S} \circ \mathcal{T}\lambda^k_{\mathcal{F}^{m+l}S} \circ \mathcal{TF}^k\mu_{\mathcal{F}^{m+l}S} \circ \mathcal{TF}^k\mathcal{T}\lambda^m_{\mathcal{F}^l S} \\ &\quad \circ \mathcal{TF}^k\mathcal{TF}^m\gamma \circ \mathcal{TF}^k\beta \circ \alpha \end{aligned}$$

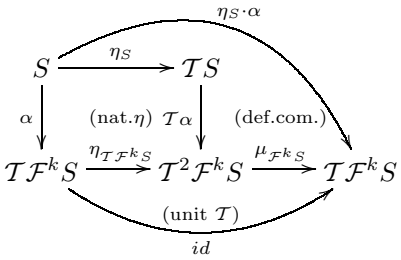
Hence, the associativity is a consequence of the commutativity of the following diagram.



(*ii*) Let $\langle S, \alpha\rangle$ be such that $\alpha : S \to \mathcal{T}\mathcal{F}^k S$. The system $\langle S, \eta_S\rangle$ is a right unit since



and it is a left unit since



□

Exponentiation of systems can also be defined, in the usual way: Let $\langle S, \alpha\rangle$ be a system of type $\mathcal{T}\mathcal{F}^n$. Then $\alpha^0 = \eta_S$ and $\alpha^{n+1} = \alpha^n \cdot \alpha$. By the associativity of the composition we have $\alpha^{n+m} = \alpha^n \cdot \alpha^m$, for any $n, m \in \mathbb{N}$.

Remark 6.3.11. We note that the definition of composition as well as Lemma 6.3.10 are related to Kleisli categories. For a monad $\mathcal{T} = \langle \mathcal{T}, \eta, \mu \rangle$, by $\mathsf{Set}_{\mathcal{T}}$ we denote the Kleisli category associated to $\mathcal{T}$. Its objects are sets and morphisms $f : X \to_{\mathcal{T}} Y$ are functions $f : X \to \mathcal{T}Y$. The identity morphism is then η_X for any set X and two morphisms $f : X \to_{\mathcal{T}} Y$ and $g : Y \to_{\mathcal{T}} Z$ compose to a morphism $g \circ_{\mathcal{T}} f : X \to_{\mathcal{T}} Z$ given by

$$g \circ_{\mathcal{T}} f = \mu_Z \circ \mathcal{T}g \circ f.$$

Given a Set endofunctor $\mathcal{F}$ with a distributive law $\lambda : \mathcal{F}\mathcal{T} \Longrightarrow \mathcal{T}\mathcal{F}$, we can lift $\mathcal{F}$ to an endofunctor $\mathcal{F}_{\mathcal{T}}$ on the Kleisli category which acts as follows

$$\mathcal{F}_{\mathcal{T}}(X) = \mathcal{F}X \qquad \mathcal{F}_{\mathcal{T}}(f) = \lambda_Y \circ \mathcal{F}f$$

for $f : X \to_{\mathcal{T}} Y$ a morphism in $\mathsf{Set}_{\mathcal{T}}$. By Lemma 6.3.7 and Lemma 6.3.8 also $\mathcal{F}^k$ lifts to a functor $\mathcal{F}^k_{\mathcal{T}}$ on the Kleisli category. Then the composition of coalgebras $\alpha \cdot \beta$ corresponds to composing some morphisms in $\mathsf{Set}_{\mathcal{T}}$ in particular $\alpha \cdot \beta = \mathcal{F}^k_{\mathcal{T}}\beta \circ_{\mathcal{T}} \alpha$. Hence, the proof of Lemma 6.3.10 could also be given via the Kleisli category. There is only an obligation to prove that $\mathcal{F}^k$ lifts in $\mathsf{Set}_{\mathcal{T}}$ to the exponent of the lifting of $\mathcal{F}$, i.e.

$$\mathcal{F}^k_{\mathcal{T}} = (\mathcal{F}_{\mathcal{T}})^k$$

which can be done by induction, by the definition of λ^k and by Lemma 6.3.8.

We next provide examples that show how the composition is defined for LTSs and for generative probabilistic systems.

Example 6.3.12. The functor defining the LTSs is of a form $\mathcal{T}\mathcal{F}$ for $\mathcal{T} = \mathcal{P}$, the powerset monad, and $\mathcal{F} = \underline{A} \times \mathcal{I}d$. By Lemma 6.3.5 the composition (and exponentiation) is defined for LTSs. Moreover, since

$$(\underline{A^k} \times \mathcal{I}d) \circ (\underline{A^m} \times \mathcal{I}d) \cong (\underline{A^{k+m}} \times \mathcal{I}d) \tag{6.16}$$

if $\langle S, \alpha : S \to \mathcal{P}(\underline{A^k} \times \mathcal{I}d) \rangle$ and $\langle S, \beta : S \to \mathcal{P}(\underline{A^m} \times \mathcal{I}d) \rangle$, then $\langle S, \alpha \cdot \beta : S \to \mathcal{P}(\underline{A^{k+m}} \times \mathcal{I}d) \rangle$. Some derivations suffice to see that

$$(\alpha \cdot \beta)(s) = \{\langle uv, t \rangle \mid \exists r \in S : \langle u, r \rangle \in \alpha(s), \langle v, t \rangle \in \beta(r)\}$$

i.e., $s \xrightarrow{uv} t$ in $\langle S, \alpha \cdot \beta \rangle$ if and only if there exists $r \in S$ such that $s \xrightarrow{u} r$ in $\langle S, \alpha \rangle$ and $r \xrightarrow{v} t$ in $\langle S, \beta \rangle$, for arbitrary state $s \in S$ and arbitrary words $u \in A^k$, $v \in A^m$.

Example 6.3.13. The generative probabilistic systems are defined by the functor $\mathcal{D}(\underline{A} \times \mathcal{I}d)$ i.e., by a functor $\mathcal{T}\mathcal{F}$ for $\mathcal{F}$ as in the previous example, and $\mathcal{T} = \mathcal{D}$ being the distribution monad. Example 6.3.4 provides also a distributive law of $\mathcal{F}$ over the monad $\mathcal{T}$. Hence, composition and exponentiation of generative probabilistic systems is also defined. Using the isomorphism (6.16), and the definition of composition, after some derivations we

get that if $\langle S, \alpha : S \to \mathcal{D}(\underline{A^k} \times \mathcal{I}d)\rangle$ and $\langle S, \beta : S \to \mathcal{D}(\underline{A^m} \times \mathcal{I}d)\rangle$, then $\langle S, \alpha \cdot \beta : S \to \mathcal{D}(\underline{A^{k+m}} \times \mathcal{I}d)\rangle$ is given by

$$(\alpha \cdot \beta)(s)(uv, t) = \sum_{r \in S} \alpha(s)(u, r) \cdot \beta(r)(v, t)$$

for $u \in A^k, v \in A^m$ and $s, t \in S$.

6.3.3 Relation to traces and other semantics

The interest in compositions was already invoked when considering weak bisimulations. It was an important issue to know what does it mean to perform several consecutive steps from a state. Later, the work of Jacobs [Jac04] on trace semantics for coalgebras of type $\mathcal{PF}$ for $\mathcal{F}$ a polynomial functor drew the author's attention to monads and distributive laws and made it easy to define composition of coalgebras of type $\mathcal{TF}$ with $\mathcal{T}$ a monad and a corresponding distributive law. The author's ambition was to extend the results on traces for other coalgebras but of type $\mathcal{PF}$. The definition of composition is a small step in this direction. In a recent work, Hasuo and Jacobs [HJ05b] proposed a different treatment of (finite) traces for $\mathcal{PF}$ coalgebras. More recently, the same authors also obtained traces for coalgebras of type $\mathcal{D}_{\leq}\mathcal{F}$ [HJ05a]. The trace map in this new result is defined in terms of composition i.e. exponentiation of coalgebras. The same can be done for the trace map from [HJ05b]. The compositions are not an essential part of the results, but they do provide a nice presentation. Generalizing the traces result i.e. obtaining traces for more general coalgebras, for example of type $\mathcal{TF}$ for any monad $\mathcal{T}$, a polynomial functor $\mathcal{F}$, with a corresponding distributive law, is an interesting direction for future work.

Coming back to weak bisimulations, we believe that compositions might also help in obtaining $*$-extensions (see Chapter 5). For example, given an LTS coalgebra $\langle S, \alpha : S \to \mathcal{P}(A \times S)\rangle$, the $*$-extension $\langle S, \alpha^* : S \to \mathcal{P}(A^* \times S)\rangle$ can be expressed in terms of compositions by

$$\alpha^*(s) = \bigcup_{n \in \mathbb{N}} \alpha^n(s)$$

where α^n denotes the n-th exponent of the coalgebra $\langle S, \alpha\rangle$. There might be similar connections between exponents and $*$-extensions for general coalgebras as well.

6.4 Paths in coalgebras

In this section we investigate possible definitions of paths for coalgebras. The case of LTS makes us believe that having a good definition of paths brings possibilities of defining various semantic relations.

Assume we have a system $\langle S, \alpha \rangle$ of type $\mathcal{F}$. A (finite) path could be a sequence of transitions

$$s_0 \; \alpha(s_0) \; s_1 \; \alpha(s_1) \; s_2 \cdots s_{n-1} \; \alpha(s_{n-1}) \; s_n \tag{6.17}$$

where, each s_i is "reachable" from $\alpha(s_{i-1})$ for $i \geq 1$. In case of transition systems, i.e. the powerset functor, it is intuitively clear that reachable means "belongs to" i.e. we require $s_i \in \alpha(s_{i-1})$.

Still, the usual notions of paths for TS and LTS are *linear*, being sequences of states and actions, unlike those from (6.17). In an LTS a path is an alternating sequence of states and actions, usually represented as

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots s_{n-1} \xrightarrow{a_n} s_n.$$

Similarly, in a transition system a path is only a sequence of states

$$s_0 \rightarrow s_1 \rightarrow s_2 \cdots \rightarrow s_n.$$

Moreover, in a simple Segala system [Seg95b], a path is a sequence

$$s_0 \xrightarrow{a_1} \mu_1 \; s_1 \xrightarrow{a_2} \mu_2 \; s_2 \cdots s_{n-1} \xrightarrow{a_n} \mu_n \; s_n$$

where $s_i \in \mathrm{supp}(\mu_i)$. This definition of a path is semi-linear. On the one hand, it exploits the similarity with LTS and therefore shows linearity and, on the other hand, it involves whole distributions over states.

For generative probabilistic systems the usual notion of a path (see e.g. Chapter 5) is also linear. A path in a generative system is an alternating sequence

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots s_{n-1} \xrightarrow{a_n} s_n$$

such that the probability of performing each transition $s_{i-1} \xrightarrow{a_i} s_i$ is greater than 0, for $i \geq 1$. This is very much different than what comes out of (6.17). The advantage of the used definition is that it is indeed linear, the disadvantage is that it looses probabilistic information. The behavior of the state is no longer determined by the set of paths, nor by the set of paths of length one, in contrast to the LTS case.

In this section we will present some general observations that show, for example, the general principles that lead to this linear but incomplete definition of paths for generative systems.

Jacobs [Jac04] defines paths in a semi-linear fashion for systems of type $\mathcal{PF}$, where $\mathcal{F}$ preserves weak pullbacks, as sequences

$$\langle u_0, u_1, \ldots, u_n \rangle \in \prod_{i=0}^{n} \mathcal{F}^i S$$

such that for $i \geq 0$ we have

$$\langle u_{i+1}, u_i \rangle \in \mathrm{Rel}(\mathcal{F})^i((id \times \alpha)^{-1} \in_{\mathcal{F}S}). \tag{6.18}$$

Jacobs' definition of a path implicitly uses the existing power law and the fact that $\in_X$ is the reachability relation.

In general, let $\mathcal{T}$ be a monad and $\mathcal{F}$ a functor, with a distributive law $\lambda : \mathcal{F}\mathcal{T} \Longrightarrow \mathcal{T}\mathcal{F}$. Moreover, let a family of reachability relations be given $\mathcal{R} = \{R_X\}$ where $R_X \subseteq X \times \mathcal{T}X$ for any set X. Intuitively, this is how we form paths of $\langle S, \alpha : S \to \mathcal{T}\mathcal{F}S\rangle$ w.r.t. $\mathcal{R}$: We pick up a first element $u_0 \in S$. We apply the transition function to it and get $\alpha(u_0) \in \mathcal{T}\mathcal{F}S$. We pick up a next element $u_1 \in \mathcal{F}S$ such that $\langle u_1, \alpha(u_0)\rangle \in R_{\mathcal{F}S}$. Then we continue from u_1, we apply now $\mathcal{F}\alpha$ to it in order to get its transitions and we land in $\mathcal{F}\alpha(u_1) \in \mathcal{F}\mathcal{T}\mathcal{F}S$. An application of the distributive law gets us back on the right track, and we have $\lambda_{\mathcal{F}S}(\mathcal{F}\alpha(u_1)) \in \mathcal{T}\mathcal{F}^2S$. At this point we are in position to again pick up a next element for our path. We choose $u_2 \in \mathcal{F}^2S$ such that $\langle u_2, \lambda_{\mathcal{F}S}(\mathcal{F}\alpha(u_1))\rangle \in R_{\mathcal{F}^2S}$. Proceeding this way, we build possible paths.

Hence, paths for systems of type $\mathcal{T}\mathcal{F}$ with $\mathcal{T}$ being a monad and a corresponding distributive law, w.r.t a family $\mathcal{R}$ are sequences

$$\langle u_0, u_1, \ldots, u_n\rangle \in \prod_{i=0}^{n} \mathcal{F}^i S$$

such that for all $i = 0, \ldots, n-1$ we have

$$\langle u_{i+1}, u_i\rangle \in (id \times \lambda^i_{\mathcal{F}S}\mathcal{F}^i\alpha)^{-1} R_{\mathcal{F}^{i+1}S} = R^{\alpha,i}. \tag{6.19}$$

Remark 6.4.1. We note that Jacobs' paths are indeed paths according to (6.19) in case of the powerset monad and the membership relations. Let $\mathcal{T} = \mathcal{P}$, π^n the n-fold power law, and $R_X = \in_X$. Moreover, let $\langle S, \alpha\rangle$ be a $\mathcal{P}\mathcal{F}$ coalgebra, for a weak pullback preserving functor $\mathcal{F}$. Then

$$\mathrm{Rel}(\mathcal{F})^i(\in_{\mathcal{F}S}) = (id \times \pi^i_{\mathcal{F}S})^{-1}(\in_{\mathcal{F}^{i+1}S}) \tag{6.20}$$

as can be derived from [Jac04, Lemma 4.2]. This implies that

$$\begin{aligned}
\mathrm{Rel}(\mathcal{F})^i((id \times \alpha)^{-1}(\in_{\mathcal{F}S})) &\overset{(*)}{=} (id \times \mathcal{F}^i\alpha)^{-1}\,\mathrm{Rel}(\mathcal{F})^i(\in_{\mathcal{F}S}) \\
&\overset{(6.20)}{=} (id \times \mathcal{F}^i\alpha)^{-1}(id \times \pi^i_{\mathcal{F}S})^{-1}(\in_{\mathcal{F}^{i+1}S}) \\
&= (id \times \pi^i_{\mathcal{F}S}\mathcal{F}^i\alpha)^{-1}(\in_{\mathcal{F}^{i+1}S})
\end{aligned}$$

where the equality $(*)$ holds by the properties of relation liftings (see Section 3.6.1).

Hence, both notions of paths coincide for LTS, and they also correspond to the usual notion of paths for LTS, as shown in the next example.

Example 6.4.2. Let $\mathcal{T} = \mathcal{P}$, $\mathcal{F} = \underline{A} \times \mathcal{I}d$ and $R_X = \in_X$. Let π^n be the distributive laws from Example 6.3.9. A sequence $\langle u_0, \ldots, u_n\rangle$ is a path, $u_i \in A^i \times S$ if for all $i \geq 0$

$$\langle u_{i+1}, u_i\rangle \in (id \times \pi^i_{\mathcal{F}S}\mathcal{F}^i\alpha)^{-1} R_{\mathcal{F}^{i+1}S}.$$

i.e.

$$u_{i+1} \in \pi^i_{\mathcal{F}S}((\underline{A^i} \times \alpha)(u_i)).$$

Now, since $u_i = \langle w_i, s_i \rangle \in A^i \times S$, and $(\underline{A^i} \times \alpha)(u_i) = \langle w_i, \alpha(s_i) \rangle$, from Example 6.3.9, we get

$$\begin{aligned} \pi^i_{\mathcal{F}S}(\langle w_i, \alpha(s_i)\rangle) &= \{\langle w_i, u\rangle \mid u \in \alpha(s_i)\} \\ &= \{\langle w_i, \langle a, s'\rangle\rangle \mid \langle a, s'\rangle \in \alpha(s_i)\} \\ &\cong \{\langle w_i \cdot a, s'\rangle \mid s \xrightarrow{a} s'\}. \end{aligned}$$

Hence, $\langle u_0, \ldots, u_n \rangle$ is a path, $u_i = \langle w_i, s_i \rangle \in A^i \times S$ if and only if for all $i \geq 0$ it holds that $w_{i+1} = w_i \cdot a$ for some $a \in A$ and $s_i \xrightarrow{a} s_{i+1}$.

The definition of paths for systems of type $\mathcal{TF}$ depends on a family of relations $\mathcal{R}$. We do not know what characterizes a good family $\mathcal{R} = \{R_X \subseteq X \times \mathcal{T}X\}$ of relations for reachability. In any case, every such family of relations should satisfy the following condition. For any natural number n, any system $\langle S, \alpha \rangle$ of type $\mathcal{TF}$ and any state $s \in S$

$$(id \times \alpha^n)^{-1}(R_{\mathcal{F}^n S}) \cap \mathcal{F}^n S \times \{s\} = R^{\alpha,n} \circ \cdots \circ R^{\alpha,1} \cap \mathcal{F}^n S \times \{s\} \qquad (6.21)$$

The condition provides a link between the notion of composition of systems and the notion of a path. It can be rewritten to

$$\begin{aligned} &\{u \mid \langle u, \alpha^n(s)\rangle \in R_{\mathcal{F}^n S}\} = \\ &\qquad \{u \mid \exists u_1, \ldots, u_{n-1} \colon \langle s, u_1, \ldots, u_{n-1}, u\rangle \text{ is a path in } \langle S, \alpha\rangle \text{ w.r.t } R_X\} \end{aligned}$$

expressing that reachable elements from $\alpha^n(s)$ are exactly those that can be reached by a path of length n.

We next show that for $\mathcal{T}$ being a submonad of $\mathcal{P}$ in the sense that there exists a natural transformation $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$, families of reachability relations come naturally.

6.5 Paths for submonads of $\mathcal{P}$

Since the powerset monad, together with the family of membership relations, seems to play a special role in defining paths of systems, it makes sense to study in more detail submonads of $\mathcal{P}$ i.e. monads that can be naturally mapped to the powerset monad. In order to limit the size of this section we shall skip or present only sketches of proofs. The complete proofs can be found in [Sok05].

Lemma 6.5.1. *The following are equivalent:*

(*i*) *The monad $\mathcal{T}$ is submonad of $\mathcal{P}$, i.e. there exists a natural transformation $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$.*

(*ii*) *There exists a family* $\mathcal{R} = \{R_X \subseteq X \times \mathcal{T}X\}$ *of relations indexed by sets such that for all sets* X, Y *and all* $f : X \to Y$

$$(f \times id)R_X = (id \times \mathcal{T}f)^{-1}R_Y. \tag{6.22}$$

Proof (sketch) Assume (*i*) holds and $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$. Then we can define a family of relations $\mathcal{R} = \{R_X\}$ by

$$R_X = (id \times \sigma_X)^{-1}(\in_X) \subseteq X \times \mathcal{T}X \tag{6.23}$$

which satisfies (6.22).

For the opposite, ff $\mathcal{R} = \{R_X \subseteq X \times \mathcal{T}X\}$ is a family of relations with the property (6.22), then we define a family of maps $\{\sigma_X\}$ by

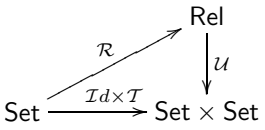
$$\sigma_X : \mathcal{T}X \to \mathcal{P}X, \quad \sigma_X(u) = \{x \in X \mid \langle x, u\rangle \in R_X\}. \tag{6.24}$$

and they form a natural transformation $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$. □

In the proof above, if σ satisfying (6.22) exists, then we call the family of relations R_X defined by (6.23), the family associated to σ. Conversely, if R_X exists satisfying (6.22), then we say that the natural transformation σ from (6.24) is associated to the family. These assignments are inverses to each other. Moreover, the condition (6.22) implies that

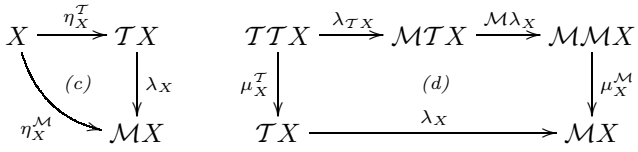
$$(f \times \mathcal{T}f)R_X \subseteq R_Y$$

which is equivalent to the condition that the family $\mathcal{R} = \{R_X\}$ is functorial, i.e. $\mathcal{R}$ determines a functor making the following diagram commute.



The next lemma shows that a family of relations associated to a submonad of $\mathcal{P}$ can be used instead of the membership family in case the submonad natural transformation is a monad map. We first define the notion of a monad map.

Definition 6.5.2. *Let* $\langle \mathcal{T}, \eta^{\mathcal{T}}, \mu^{\mathcal{T}}\rangle$ *and* $\langle \mathcal{M}, \eta^{\mathcal{M}}, \mu^{\mathcal{M}}\rangle$ *be two monads. A monad map from* $\mathcal{T}$ *to* $\mathcal{M}$ *is a natural transformation* $\lambda : \mathcal{T} \Longrightarrow \mathcal{M}$ *such that it preserves the monad structures, i.e., the following two diagrams commute.*



Lemma 6.5.3. *Let $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$ be a monad map and let $\mathcal{R}$ be the family associated to σ, i.e. $R_X = (id \times \sigma_X)^{-1}(\in_X)$. Let $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$ be a distributive law. Then $\mathcal{R}$ satisfies condition (6.21).* □

Example 6.5.4. We have a natural transformation $\mathrm{supp} : \mathcal{D} \Longrightarrow \mathcal{P}$ mapping any distribution to its support set, i.e., $\mathrm{supp}_X(\mu) = \mathrm{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$ for any $\mu \in \mathcal{D}X$. The associated family of relations with the property (6.22), by (6.23) is

$$\begin{aligned} R_X &= (id \times \mathrm{supp}_X)^{-1}(\in_X) \\ &= \{\langle x, \mu\rangle \mid \langle x, \mathrm{supp}(\mu)\rangle \in \in_X\} \\ &= \{\langle x, \mu\rangle \mid x \in \mathrm{supp}(\mu)\}. \end{aligned}$$

We now show that, under a reasonable assumption, if $\mathcal{R}$ is the family of relations associated to a natural transformation $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$, then one can define paths via (6.19) or via (6.18) with $\mathcal{R}$ instead of $\in$ obtaining the same notion. For this we introduce first the notion of a *map of distributive laws*. We say that $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$ is a map of the distributive laws λ and π, notation $\sigma : \lambda \Longrightarrow \pi$, if the next diagram commutes.

$$\begin{array}{ccc} \mathcal{FT}X & \xrightarrow{\mathcal{F}\sigma_X} & \mathcal{FP}X \\ {\scriptstyle \lambda_X}\downarrow & & \downarrow{\scriptstyle \pi_X} \\ \mathcal{TF}X & \xrightarrow{\sigma_{\mathcal{F}X}} & \mathcal{PF}X \end{array} \tag{6.25}$$

In a sense, a map of distributive laws shows that two distributive laws are compatible, or imitate each other along the natural transformation σ.

Lemma 6.5.5. *Let $\mathcal{T}$ be a monad, $\mathcal{F}$ a functor, $\lambda : \mathcal{FT} \Longrightarrow \mathcal{TF}$ a (plain) distributive law, and $\pi : \mathcal{FP} \Longrightarrow \mathcal{PF}$ the power law. Moreover, let $\sigma : \lambda \Longrightarrow \pi$ and let $\mathcal{R} = \{R_X\}$ be the associated family of relations. Then*

$$(id \times \lambda^i_X)^{-1}(R_{\mathcal{F}^i X}) = \mathrm{Rel}(\mathcal{F})^i(R_X). \tag{6.26}$$

□

It can easily be seen, as in Remark 6.4.1, that Equation (6.26) implies equivalence of the Conditions (6.19) and (6.18).

Maps between distributive laws exist in the nature, as the following example demonstrates.

Example 6.5.6. Let $\lambda : \underline{A} \times \mathcal{D} \Longrightarrow \mathcal{D}(\underline{A} \times \mathcal{I}d)$ and $\pi : \underline{A} \times \mathcal{P} \Longrightarrow \mathcal{P}(\underline{A} \times \mathcal{I}d)$ be defined as in Example 6.3.4 and Example 6.3.6, respectively. Consider the support natural transformation $\mathrm{supp} : \mathcal{D} \Longrightarrow \mathcal{P}$. Then supp is a map of distributive laws, $\mathrm{supp} : \lambda \Longrightarrow \pi$, since one can directly verify that

$$\pi_X \circ \mathcal{F}\,\mathrm{supp}_X = \mathrm{supp}_{\mathcal{F}X} \circ \lambda_X.$$

Hence, for generative probabilistic systems there is one notion of a path with respect to the support relations. Moreover, it can be seen that this one notion corresponds to the usual linear notion of a path for generative systems.

The following result suggests that in the case of the powerset monad, the family of membership relations deserves to be called the family of reachability relations.

Lemma 6.5.7. *There exist exactly two families of relations R_X that satisfy (6.22), associated to the powerset monad $\mathcal{P}$. These are $R_X = \emptyset$ for all sets X, and $R_X = \in_X$.* □

We next point that for any submonad of $\mathcal{P}$ there is a largest natural transformation that witnesses the submonad property. It corresponds to a largest family of relations. First we order the families of relations and the natural transformations. Let $\mathcal{R} = \{R_X \subseteq X \times \mathcal{T}X\}$ and $\mathcal{Q} = \{Q_X \subseteq X \times \mathcal{T}X\}$. We define

$$\mathcal{R} \leq \mathcal{Q} \quad \Longleftrightarrow \quad R_X \subseteq Q_X$$

for all sets X. Furthermore, let $\lambda : \mathcal{T} \Longrightarrow \mathcal{P}$ and $\tau : \mathcal{T} \Longrightarrow \mathcal{P}$. Define

$$\lambda \leq \tau \iff \lambda_X(u) \subseteq \tau_X(u)$$

for all sets X and all $u \in \mathcal{T}X$. One directly verifies that if $\mathcal{R}, \mathcal{Q}$ are the families of relations associated to the natural transformations λ, τ from $\mathcal{T}$ to $\mathcal{P}$, respectively, then

$$\mathcal{R} \leq \mathcal{Q} \quad \Longleftrightarrow \quad \lambda \leq \tau.$$

For any monad $\mathcal{T}$, there exists the empty natural transformation $\varepsilon : \mathcal{T} \Longrightarrow \mathcal{P}$ given by $\varepsilon_X(u) = \emptyset$ for all sets X and all $u \in \mathcal{T}X$. Furthermore, if $\{\mathcal{R}^i \mid i \in I\}$ is a collection of families of relations that satisfy (6.22), then $\mathcal{R}$ with components $R_X = \bigcup_{i \in I} R^i_X$ also satisfies (6.22), and $\mathcal{R}^i \leq \mathcal{R}$ for all $i \in I$. As a consequence we get the following property.

Lemma 6.5.8. *For any monad $\mathcal{T}$, there exists a largest family of relations $\mathcal{R} = \{R_X \subseteq X \times \mathcal{T}X\}$ with the property (6.22), and a corresponding largest natural transformation $\sigma : \mathcal{T} \Longrightarrow \mathcal{P}$.* □

Example 6.5.9. The family $\mathcal{R}$ corresponding to the support natural transformation $\mathrm{supp} : \mathcal{D} \Longrightarrow \mathcal{P}$ is the largest family of relations that satisfies (6.22).

In the last two sections we studied ways to define paths in coalgebras. We are not yet convinced whether it is reasonable to define notions of linear behavior, such as paths, for general coalgebras. The most general definition given by condition (6.17) does not seem to reflect intuition of what a linear path should be. Moreover, there is the question of how to pick next states, i.e. which states are "reachable" from a transition $\alpha(s)$.

Therefore, we discussed subclasses of coalgebras for which a definition of a path is possible. Such are the coalgebras of a monad, the coalgebras of type

$\mathcal{PF}$ [Jac04], and, as we have seen, also coalgebras of type $\mathcal{TF}$ for $\mathcal{T}$ being a submonad of $\mathcal{P}$. An example of such coalgebras are the generative probabilistic systems, and for them we obtain the usual notion of a path. While studying the possibility of defining paths, we have come to some interesting observations for the submonads of $\mathcal{P}$. Still, we are not convinced that defining paths by "forgetting" parts of the behavior, as in the case of generative systems is a good idea. Application of these notions of paths for obtaining semantic relations remains an issue for future research. It could be a way to evaluate the notions of paths that we have discussed.

Moreover, a notion of paths plays a significant role in the definition of open maps bisimulation [FCW99] where bisimulation is characterized by a category of paths. We leave the investigation of the connection between open maps and our notions of paths for future work.

Bibliography

[AB02] S. Andova and J.C.M. Baeten, *Alternative composition does not imply non-determinism*, Bulletin of the European Association for Theoretical Computer Science **76** (2002), 125–127.

[AH99] R. Alur and T.A. Henzinger, *Reactive modules*, Formal Methods in System Design **15** (1999), 7–48, A preliminary version appeared in the Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press, 1996, pp. 207-218.

[AHJ01] L. de Alfaro, T.A. Henzinger, and R. Jhala, *Compositional methods for probabilistic systems*, CONCUR 2001 - Concurrency Theory: 12th International Conference, Aalborg, Denmark, August 20-25, 2001, LNCS, vol. 2154, Springer-Verlag, 2001, pp. 351–365.

[Alf97] L. de Alfaro, *Formal verification of probabilistic systems*, Ph.D. thesis, Stanford University, 1997.

[Alf98] _____, *Stochastic transition systems*, International Conference on Concurrency Theory, CONCUR, vol. 1466, LNCS, 1998, pp. 423–438.

[AM89] Peter Aczel and Nax Mendler, *A final coalgebra theorem*, Proc. 3rd CTCS (D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, eds.), LNCS, vol. 389, Springer, 1989, pp. 357–365.

[And99] S. Andova, *Process algebra with probabilistic choice*, Proc. 5th International AMAST Workshop, ARTS'99, Bamberg, Germany (J.-P. Katoen, ed.), LNCS 1601, Springer-Verlag, 1999, pp. 111–129.

[And02] _____, *Probabilistic process algebra*, Ph.D. thesis, Eindhoven University of Technology, 2002.

[AW05] S. Andova and T.A.C. Willemse, *Equivalences for silent transitions in probabilistic systems*, EXPRESS'04, ENTCS 128(2), 2005, Full version will appear in TCS, pp. 53–66.

[BA95] A. Bianco and L. de Alfaro, *Model checking of probabilistic and nondeterministic systems*, Found. of Software Tech. and Theor. Comp. Sci., LNCS, vol. 1026, Springer-Verlag, 1995.

[Bai96] C. Baier, *Polynomial time algorithms for testing probabilistic bisimulation and simulation*, Proc. 8th International Conference on Computer Aided Verification (CAV'96), Lecture Notes in Computer Science, vol. 1102, 1996, pp. 38–49.

[Bai98] C. Baier, *On algorithmic verification methods for probabilistic systems*, Habilitationsschrift, FMI, Universitaet Mannheim, 1998.

[Bal01] G. Balbo, *Introduction to stochastic petri nets*, Lectures on Formal Methods and Performance Analysis, First EEF/Euro Summer School on Trends in Computer Science, Berg en Dal, The Netherlands, July 3-7, 2000 (E. Brinksma, H. Hermanns, and J.-P. Katoen, eds.), LNCS, vol. 2090, Springer-Verlag, 2001, pp. 84–155.

[Bar04] F. Bartels, *On generalised coinduction and probabilistic specification formats: distributive laws in coalgebraic modelling*, Ph.D. thesis, Vrije Universiteit, Amsterdam, 2004.

[BBS95] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka, *Axiomatizing probabilistic processes: ACP with generative probabilities*, Information and Computation **121** (1995), no. 2, 234–255.

[BDEP97] R. Blute, J. Desharnais, A. Edalat, and P. Panangaden, *Bisimulation for labelled Markov processes*, LICS'97, 1997, pp. 149–158.

[BDHK99] C. Baier, P.R. D'Argenio, H. Hermanns, and J.-P. Katoen, *How to cook a probabilistic process calculus*, unpublished, 1999.

[BEMC99] C. Baier, B. Engelen, and M. Majster-Cederbaum, *Deciding bisimilarity and similarity for probabilistic processes*, Journal of Computer and System Sciences **60** (1999), 187–231.

[Ber99] M. Bernardo, *Theory and application of extended Markovian process algebra*, Ph.D. thesis, University of Bologna, 1999.

[BG98] M. Bernardo and R. Gorrieri, *A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time*, Theoretical Computer Science **202** (1998), no. 1, 1–54.

[BH97] C. Baier and H. Hermanns, *Weak bisimulation for fully probabilistic processes*, Proc. CAV'97 (O. Grumberg, ed.), LNCS 1254, 1997, pp. 119–130.

[BH99] ———, *Weak bisimulation for fully probabilistic processes*, Tech. Report TR–CTIT–12, 1999.

[BH01] E. Brinksma and H. Hermanns, *Process algebra and Markov chains*, Lectures on Formal Methods and Performance Analysis, First EEF/Euro Summer School on Trends in Computer Science, Berg en Dal, The Netherlands, July 3-7, 2000 (E. Brinksma, H. Hermanns, and J.-P. Katoen, eds.), LNCS, vol. 2090, Springer-Verlag, 2001, pp. 183–232.

[BHH+04] C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle (eds.), *Validation of stochastic systems: A guide to current research*, LNCS, vol. 2925, Springer, 2004.

[BHHK00] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, *Model checking continuous-time Markov chains by transient analysis*, CAV 2000, vol. 1855, LNCS, Springer-Verlag, 2000, pp. 358–372.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe, *A theory of communicating sequential processes*, Journal of the ACM **31** (1984), 560–599.

[BK85] J.A. Bergstra and J.W. Klop, *Algebra of communicating processes with abstraction*, Theoretical Computer Science **37** (1985), 77–121.

[BK97] C. Baier and M.Z. Kwiatkowska, *Domain equations for probabilistic processes*, 4th Workshop on Expressiveness in Concurrency (EXPRESS'97), Santa Margherita, vol. 7, Electronic Notes in Theoretical Computer Science, 1997.

[BK00] _____, *Domain equations for probabilistic processes*, Mathematical Structures in Computer Science **10** (2000), 665–717.

[BLFG95] A. Benveniste, B. C. Levy, E. Fabre, and P. Le Guernic, *A calculus of stochastic systems for the specification, simulation, and hidden state estimation of mixed stochastic/non-stochastic systems*, Theoretical Computer Science **152** (1995), 171–217.

[BM89] B. Bloom and A. R. Meyer, *A remark on bisimulation between probabilistic processes*, Foundations of Software Technology and Theoretical Computer Science, LNCS, vol. 363, Springer-Verlag, 1989, pp. 26–40.

[Bor94] F. Borceux, *Handbook of categorial algebra*, Cambridge University Press, 1994.

[BS00] C. Baier and M.I.A. Stoelinga, *Norm fuctions for probabilistic bisimulations with delays*, Proceedings of 3rd International Conference on Foundations of Science and Computation Structures (FOSSACS), Berlin, Germany, March 2000 (J. Tiuryn, ed.), LNCS, vol. 1784, "Springer - verlag", 2000, pp. 1–16.

[BS01] E. Bandini and R. Segala, *Axiomatizations for probabilistic bisimulation*, Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP) 2001, Crete, LNCS 2076, 2001, pp. 370–381.

[BSV03] F. Bartels, A. Sokolova, and E.P. de Vink, *A hierarchy of probabilistic system types*, Electronic Notes in Theoretical Computer Science (H. Peter Gumm, ed.), vol. 82, Elsevier, 2003.

[BSV04] ———, *A hierarchy of probabilistic system types*, Theoretical Computer Science **327** (2004), 3–22.

[Buc94] P. Buchholz, *Markovian process algebra: Composition and equivalence*, in Proc. of PAPM '94, Erlangen (Germany), 1994, pp. 11–30.

[BW90] J.C.M. Baeten and W.P. Weijland, *Process algebra*, Cambridge University Press, 1990.

[CC91] L. Christoff and I. Christoff, *Efficient algorithms for verification of equivalences for probabilistic processes*, Proc. Workshop on Computer Aided Verification 1991 (K. Larsen and A. Skou, eds.), LNCS, vol. 575, 1991.

[CC04] D.R. Cacciagrano and F. Corradini, *Expressiveness of timed events and timed languages*, Formal Methods for the Design of Real-Time Systems (M. Bernardo and F. Corradini, eds.), LNCS 3185, 2004, pp. 98–131.

[CH05] L. Cheung and M. Hendriks, *Casual dependencies in parallel composition of stochastic processes*, 2005, under review. Available via: `http://www.niii.ru.nl/~lcheung/`.

[Chr90] I. Christoff, *Testing equivalences and fully abstract models for probabilistic processes*, Proceedings of CONCUR'90 (J.C.M. Baeten and J.W. Klop, eds.), LNCS 458, Springer-Verlag, 1990, pp. 126–140.

[CSZ92] R. Cleaveland, S.A. Smolka, and A. Zwarico, *Testing preorders for probabilistic processes*, Automata, Languages and Programming (ICALP '92), Vienna, LNCS, vol. 623, Springer-Verlag, 1992, pp. 708–719.

[CY95] C. Courcoubetis and M. Yannakakis, *The complexity of probabilistic verification*, Journal of the ACM (JACM) **42** (1995), 857–907.

[D'A99] P.R. D'Argenio, *Algebras and automata for timed and stochastic system*, Ph.D. thesis, University of Twente, 1999.

[DEP98] J. Desharnais, A. Edalat, and P. Panangaden, *A logical characterization of bisimulation for labeled Markov processes*, Proc. LICS'98 (Indianapolis), 1998, pp. 478–487.

[DEP02] ———, *Bisimulation for Labelled Markov Processes*, Information and Computation **179** (2002), 163–193.

[Der70] C. Derman, *Finite state Markovian decision proceses*, Academic Press, 1970.

[DGJP02a] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, *The metric analogue of weak bisimulation for probabilistic processes*, Proc. LICS 2002, IEEE, 2002, pp. 413–422.

[DGJP02b] ———, *Weak bisimulation is sound and complete for PCTL*, Proc. CONCUR 2002 (L. Brim, P.Janar, M. Ketínský, and A.Kuera, eds.), LNCS 2421, 2002, pp. 355–370.

[DHK98] P. D'Argenio, H. Hermanns, and J.-P. Katoen, *On generative parallel composition*, Proc. PROBMIV'98, ENTCS 22, 1998, pp. 105–122.

[DJJL01] P.R. D'Argenio, B. Jeannet, H.E. Jensen, and K.G. Larsen, *Reachability analysis of probabilistic systems by successive refinements*, PAPM-PROBMIV 2001, Aachen, Germany (L. de Alfaro and S. Gilmore, eds.), LNCS, vol. 2165, Springer-Verlag, 2001, pp. 29–56.

[DJJL02] ———, *Reduction and refinement strategies for probabilistic analysis*, PAPM-PROBMIV 2002, Copenhagen, Denmark (H. Hermanns and R. Segala, eds.), LNCS, Springer-Verlag, 2002.

[FCW99] M. Fiore, G.L. Cattani, and G. Winskel, *Weak bisimulation and open maps*, Proc. LICS'99, IEEE, 1999, pp. 67–76.

[GJS90] A. Giacalone, C. Jou, and S. Smolka, *Algebraic reasoning for probabilistic concurrent systems*, Proc. of the Working Conf. on Programming Concepts and Methods, 1990. (M. Broy and C.B. Jones, eds.), North Holland, 1990, pp. 443–458.

[Gla90] R.J. van Glabbeek, *The linear time – branching time spectrum (extended abstract)*, Proceedings CONCUR '90, Theories of Concurrency: Unification and Extension, Amsterdam, August 1990 (J.C.M. Baeten and J.W. Klop, eds.), vol. 458, 1990, pp. 278–297.

[Gla93] ———, *The linear time – branching time spectrum II; the semantics of sequential systems with silent moves (extended abstract)*, Proceedings CONCUR'93, 4^{th} International Conference on *Concurrency Theory*, Hildesheim, Germany, August 1993 (E. Best, ed.), vol. 715, 1993, pp. 66–81.

[Gla01] ———, *The linear time – branching time spectrum I; the semantics of concrete, sequential processes*, Handbook of Process Algebra (J.A. Bergstra, A. Ponse, and S.A. Smolka, eds.), Elsevier, 2001, pp. 3–99.

[GS00] H. Peter Gumm and Tobias Schröder, *Coalgebraic structure from weak pullback preserving fuctors*, Proc. CMCS 2000 (Horst Reichel, ed.), ENTCS, vol. 33, Elsevier, 2000.

[GSS95] R.J. van Glabbeek, S.A. Smolka, and B. Steffen, *Reactive, generative, and stratified models of probabilistic processes*, Information and Computation **121** (1995), 59–80.

[GSST90] R. J. van Glabbeek, S. A. Smolka, B. Steffen, and C. M. N. Tofts, *Reactive, generative, and stratified models of probabilistic processes*, Logic in Computer Science, 1990, pp. 130–141.

[Gum99] H.P. Gumm, *Elements of the general theory of coalgebras*, Tech. Report LUATS'99, Rand Afrikaans University, 1999.

[Gum01] H. Peter Gumm, *Functors for coalgebras*, Algebra Universalis **45** (2001), 135–147.

[GW96] R.J. van Glabbeek and W.P. Weijland, *Branching time and abstraction in bisimulation semantics*, Journal of the ACM **43** (1996), no. 3, 555–600.

[Hal50] P.R. Halmos, *Measure theory*, Van Nostrand, 1950.

[Han91] H. A. Hansson, *Time and probability in formal design of distributed systems*, Ph.D. thesis, Uppsala University, Department of Computer Systems, 1991.

[Han94] H. Hansson, *Time and probability in formal design of distributed systems*, Real-Time Safety Critical Systems, vol. 1, Elsevier, 1994.

[Har02] J.I. den Hartog, *Probabilistic extensions of semantical models*, Ph.D. thesis, Vrije Universiteit Amsterdam, 2002.

[Hav01] B. R. Haverkort, *Markovian models for performance and dependability evaluation*, Lectures on Formal Methods and Performance Analysis, First EEF/Euro Summer School on Trends in Computer Science, Berg en Dal, The Netherlands, July 3-7, 2000 (E. Brinksma, H. Hermanns, and J.-P. Katoen, eds.), LNCS, vol. 2090, Springer-Verlag, 2001, pp. 38–84.

[Her98] H. Hermanns, *Interactive Markov chains*, Ph.D. thesis, Universiät Erlangen-Nürnberg, 1998, Revised version appeared as Interactive Markov Chains And the Quest for Quantified Quality, LNCS 2428, 2002.

[Hil94] J. Hillston, *A compositional approach to performance modelling*, Ph.D. thesis, University of Edinburgh, 1994, Also appeared in the CPHC/BCS Distinguished Dissertation Series, Cambridge University Press, 1996.

[HJ94] H. Hansson and B. Jonsson, *A logic for reasoning about time and reliability*, Formal Aspects of Computing **6** (1994), 512–535.

[HJ98] C. Hermida and B. Jacobs, *Structural induction and coinduction in a fibrational setting*, Information and Computation **145** (1998), 107–152.

[HJ05a] I. Hasuo and B. Jacobs, *Coalgebraic trace semantics for probabilistic systems*, CALCO-jnr Workshop, 2005.

[HJ05b] ———, *Context-free languages via coalgebraic trace semantics*, CALCO 2005, LNCS, vol. 3629, Springer, 2005, pp. 204–233.

[Hoa85] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.

[How71] R. A. Howard, *Dynamic probabilistic systems*, John Wiley & Sons, Inc., New York, 1971.

[HV98] J.I. den Hartog and E.P. de Vink, *Mixing up nondeterminism and probability: A preliminary report*, Proc. PROBMIV'98 (C. Baier, M. Huth, M. Kwiatkowska, and M. Ryan, eds.), ENTCS 22, 1998.

[HV02] ———, *Verifying probabilistic programs using a Hoare-like logic*, International Journal of Foundations of Computer Science **13** (2002), 315–340.

[Jac02] Bart Jacobs, *Exercises in coalgebraic specification*, Algebraic and coalgebraic methods in the mathematics of program construction (R. Backhouse, R. Crole, and J. Gibbons, eds.), LNCS, vol. 2297, Springer, 2002, pp. 237–281.

[Jac04] B. Jacobs, *Trace semantics for coalgebras*, Proceedings of CMCS'04, ENTCS (Jiri Adamek, ed.), Elsevier, 2004.

[Jac05] ———, *Introduction to coalgebra. towards mathematics of states and observations*, 2005, Book in preparation, draft available via `http://www.cs.ru.nl/~bart`.

[JH03] B. Jacobs and J. Hughes, *Simulations in coalgebra*, Electronic Notes in Theoretical Computer Science (H. Peter Gumm, ed.), vol. 82, Elsevier, 2003.

[JL91] B. Jonsson and K.G. Larsen, *Specification and refinement of probabilistic processes*, Proceedings of Sixth Annual IEEE Symposium on Logic in Computer Science, 1991. LICS '91., IEEE, 1991.

[JLY01] B. Jonsson, K.G. Larsen, and W. Yi, *Probabilistic extensions of process algebras*, Handbook of Process Algebras, Elsevier, North Holland, 2001.

[Jon89] C. Jones, *Probabilistic non-determinism*, Ph.D. thesis, University of Edinburgh, 1989.

[JR96] B.P.F. Jacobs and J.J.M.M. Rutten, *A tutorial on (co)algebras and (co)induction*, Bulletin of the EATCS **62** (1996), 222–259.

[JS90] C.-C. Jou and S.A. Smolka, *Equivalences, congruences and complete axiomatizations for probabilistic processes*, Proceedings of CONCUR'90 (J.C.M. Baeten and J.W. Klop, eds.), Springer-Verlag, 1990, pp. 367–383.

[JY02] B. Jonnson and W. Yi, *Testing preorders for probabilistic processes can be characterized by simulations*, Theoretical Computer Science **282** (2002), 33–51.

[KN98] M.Z. Kwiatkowska and G.J. Norman, *A testing equivalence for reactive probabilistic processes*, EXPRESS '98 Fifth International Workshop on Expressiveness in Concurrency, ENTCS 16(2), 1998.

[KS76] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Springer-Verlag, New York, 1976.

[Kur00] A. Kurz, *Logics for coalgebras and applications to computer science*, Ph.D. thesis, Ludwig-Maximilians-Universität München, 2000.

[LN04] N. López and M. Núñez, *An overview of probabilistic process algebras and their equivalences*, Validation of Stochastic Systems: A Guide to Current Research (C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle, eds.), LNCS 2925, 2004, pp. 89–123.

[Low95] G. Lowe, *Probabilistic and prioritized models of timed CSP*, Theoretical Computer Science **138** (1995), 315–352.

[LP81] L. R. Lewis and C. H. Papadimitriou, *Elements of the theory of computation*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[LS91] K. G. Larsen and A. Skou, *Bisimulation through probabilistic testing*, Information and Computation **94** (1991), 1–28.

[LS92] K.G. Larsen and A. Skou, *Compositional verification of probabilistic processes*, CONCUR '92, Third International Conference on Concurrency Theory, Stony Brook, NY, USA (R. Cleaveland, ed.), LNCS, vol. 630, Springer-Verlag, 1992, pp. 456–471.

[LT87] N. A. Lynch and M. Tuttle, *Hierarchical completeness proofs for distributed algorithms*, Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing, 1987.

[Mac71] S. MacLane, *Categories for the working mathematician*, Springer-Verlag, 1971.

[Mil80] R. Milner, *A calculus of communicating systems*, LCNS 92, 1980.

[Mil83] ———, *Calculi for synchrony and asynchrony*, Theoretical Computer Science **25** (1983), 267–310.

[Mil89] ———, *Communication and Concurrency*, Prentice-Hall, 1989.

[Mil90] ———, *Operational and algebraic semantics of concurrent processes*, Handbook of Theoretical Computer Science (J. van Leeuwen, ed.), Elsevier and MIT Press, 1990, pp. 1201–1242.

[MMSS96] C. Morgan, A. McIver, K. Seidel, and J.W. Sanders, *Refinement oriented probability for CSP*, Formal aspects of computing **8** (1996), 617–647.

[Mos99] L.S. Moss, *Coalgebraic logic*, Annals of Pure and Applied Logic **96** (1999), 277–317.

[Nor97] G. Norman, *Metric semantics for reactive probabilistic processes*, Ph.D. thesis, School of Computer Science, University of Birmingham, 1997.

[PA91] B. Plateau and K. Atif, *Stochastic automata network for modeling parallel systems*, IEEE Trans. on Software Engineering **17** (1991), 1093–1108.

[Par81] David Park, *Concurrency and automata on infinite sequences*, Theoretical Computer Science: 5th GI-Conference, Karlsruhe (P. Deussen, ed.), LNCS, vol. 104, Springer-Verlag, 1981, pp. 167–183.

[Pat03] Dirk Pattinson, *An introduction to the theory of coalgebras*, NASSLLI'03, 2003, notes accompanying the a course.

[Plo81] G.D. Plotkin, *A structural approach to operational semantics*, Tech. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[PLS00] A. Philippou, I. Lee, and O. Sokolsky, *Weak bisimulation for probabilistic systems*, Proc. CONCUR 2000 (C. Palamidessi, ed.), LNCS 1877, 2000, pp. 334–349.

[PZ86] A. Pnueli and L. Zuck, *Verification of multiprocess probabilistic protocols*, Distributed Computing **1** (1986), no. 1, 53–72.

[PZ93] ———, *Probabilistic verification*, Information and Computation **103** (1993), 1–29.

[Rab63] M.O. Rabin, *Probabilistic automata*, Information and Control **6** (1963), 230–245.

[RM02] J. Rothe and D. Mašulović, *Towards weak bisimulation for coalgebras*, Proc. Categorical Methods for Concurrency, Intercation and Mobility (A. Kurz, ed.), ENTCS 68, 2002, 15pp.

[Rot02] J. Rothe, *A syntactical approach to weak (bi)-simulation for coalgebras*, Proc. CMCS 2002, ENTCS 65, 2002, 16pp.

[RT93] J.J.M.M. Rutten and D. Turi, *Initial algebra and final coalgebra semantics for concurrency*, REX School/Symposium, LNCS 666, 1993, pp. 530–582.

[Rut96] J.J.M.M. Rutten, *Universal coalgebra: A theory of systems*, Tech. Report CS-R9652, CWI Amsterdam, 1996.

[Rut98] J. Rutten, *Relators and metric bisimulations*, Proc. CMCS'98 (B. Jacobs, L. Moss, H. Reichel, and J. Rutten, eds.), ENTCS 11, 1998, p. 8pp.

[Rut99] J.J.M.M. Rutten, *A note on coinduction and weak bisimilarity for while programs*, Theoretical Informatics and Applications (RAIRO) **33** (1999), 393–400.

[Rut00] _____, *Universal coalgebra: A theory of systems*, Theoretical Computer Science **249** (2000), 3–80.

[SCS03] E.W. Stark, R. Cleaveland, and S.A. Smolka, *A process-algebraic language for probabilistic I/O automata*, Proc. CONCUR'03 (R. Amadio and D. Lugiez, eds.), LNCS, vol. 2761, Springer, 2003, pp. 193–207.

[Seg95a] R. Segala, *A compositional trace-based semantics for probabilistic automata*, Proc. CONCUR'95, LNCS, vol. 962, Springer, 1995, pp. 234–248.

[Seg95b] _____, *Modeling and verification of randomized distributed real-time systems*, Ph.D. thesis, MIT, 1995.

[Sei95] K. Seidel, *Probabilistic communicating processes*, Theoretical Computer Science **152** (1995), 219–249.

[SL94] R. Segala and N.A. Lynch, *Probabilistic simulations for probabilistic processes*, Proc. Concur'94, LNCS 836, 1994, pp. 481–496.

[Sok05] A. Sokolova, *On compositions and paths for coalgebras*, Tech. Report CSR–05–26, TU Eindhoven, 2005, Available via: `http://www.win.tue.nl/~ana/`.

[SS90] S. A. Smolka and B.U. Steffen, *Priority as extremal probability*, Proceedings of CONCUR'90 (J.C.M. Baeten and J.W. Klop, eds.), LNCS, vol. 458, Springer-Verlag, 1990, pp. 456–466.

[ST05] R. Segala and A. Turrini, *Comparative analysis of bisimulation relations on alternating and non-alternating probabilistic models*, QEST'05, 2005, to appear.

[Sto02a] M.I.A. Stoelinga, *Alea jacta est: verification of probabilistic, real-time and parametric systems*, Ph.D. thesis, University of Nijmegen, the Netherlands, 2002.

[Sto02b] ———, *An introduction to probabilistic automata*, EATCS bulletin, vol. 78, 2002.

[SV99] M.I.A. Stoelinga and F.W. Vaandrager, *Root contention in IEEE 1394*, Proc. 5th International AMAST Workshop, ARTS'99, Bamberg, Germany (J.-P. Katoen, ed.), LNCS, vol. 1601, Springer-Verlag, 1999, pp. 53–75.

[SV03] ———, *A testing scenario for probabilistic automata*, Proceedings of the 30th International colloquium on automata, languages and programming (ICALP'03) Eindhoven, the Netherlands, June 2003, LNCS, vol. 2719, Springer-verlag, 2003, pp. 464–477.

[SV04] A. Sokolova and E.P. de Vink, *Probabilistic automata: system types, parallel composition and comparison*, Validation of Stochastic Systems: A Guide to Current Research (C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle, eds.), LNCS 2925, 2004, pp. 1–43.

[SVW04] A. Sokolova, E.P. de Vink, and H. Woracek, *Weak bisimulation for action-type systems*, Tech. Report CSR–04–16, TU Eindhoven, 2004, Available via: `http://www.win.tue.nl/~ana/`.

[SVW05] ———, *Weak bisimulation for action-type systems*, Proc. CTCS'04, ENTCS 122, 2005, pp. 211–228.

[Var85] M.Y. Vardi, *Automatic verification of probabilistic concurrent finite state programs*, Proc. FOCS'85 (Portland, Oregon), IEEE Computer Society Press, 1985, pp. 327–338.

[Vin98] E.P. de Vink, *On a functor for probabilistic bisimulation and the preservation of weak pullbacks*, Tech. Report IR–444, Vrije Universiteit Amsterdam, 1998.

[VR99] E.P. de Vink and J.J.M.M. Rutten, *Bisimulation for probabilistic transition systems: a coalgebraic approach*, Theoretical Computer Science **221** (1999), 271–293.

[Wol00] U. Wolter, *On corelations, cokernels, and coequations*, Proc. CMCS 2000 (H. Reichel, ed.), ENTCS 33, 2000, 20pp.

[Wor00] J. Worell, *Coinduction for recursive data types: partial orders, metric spaces and ω-categories*, Proc. CMCS 2000 (H. Reichel, ed.), ENTCS 33, 2000, 20pp.

[WSS97] S.-H. Wu, S. A. Smolka, and E. W. Stark, *Composition and behaviors of probabilistic I/O automata*, Theoretical Computer Science **176** (1997), 1–38.

[YL92] W. Yi and K.G. Larsen, *Testing preorders for probabilistic and nondeterministic processes*, Protocol Specification, Testing and Verification (Florida, USA), vol. 12, 1992, pp. 47–61.

[Zaa58] A.C. Zaanen, *An introduction to the theory of integration*, North-Holland, 1958.

Index of subjects

Samenvatting (Dutch summary)

Dit proefschrift verzamelt verscheidene theoretische resultaten, waardoor twee onderzoeksgebieden in de theoretische informatica verbonden worden: de theorie van coalgebra's en het probabilistisch modelleren voor verificatie doeleinden. De theorie van coalgebra's, enerzijds, biedt een generiek, categorisch, abstract raamwerk dat vele concrete noties van transitiesystemen afdekt. Anderzijds biedt de bestaande literatuur over probabilistisch modelleren een grote verscheidenheid aan probabilistische typen transitiesystemen.

Daar, waar transitiesystemen beschouwd worden als modellen van programma's, processen of systemen, teneinde te kunnen redeneren over het gedrag van het systeem in een toestand, is een notie van toestandsequivalentie nodig. Een algemeen bekende gedragsequivalentie is bisimilariteit, gebaseerd op de notie van bisimulatie. Een bisimulatie is een relatie op de toestandsruimte van transitiesystemen, die toestanden met hetzelfde stap-voor-stap gedrag identificeert. Een generieke notie van bisimulatie is één van de belangrijke bijdragen van de theorie van coalgebra's. Verschillende concrete noties van bisimulatie bestaan voor de verscheidene probabilistische transitiesystemen.

Een wezenlijk deel van dit proefschrift is gewijd aan een gedetailleerde studie naar en vergelijking van de bestaande concrete typen probabilistische systemen. De uitdrukkingskracht van deze verschillende transitiesystemen met betrekking tot de semantiek van bisimulatie wordt vergeleken. Een type wordt hoogstens zo expressief als een ander type beschouwd, als er een manier is om ieder systeem van het eerste type te transformeren tot een systeem van het tweede type, op een wijze zodat twee toestanden bisimilair zijn in het getransformeerde systeem dan en slechts dan als ze bisimilair zijn in het originele systeem. Op deze wijze kunnen de verschillende typen probabilistische systemen geordend worden in een hierarchie van expressiviteit. Voor de presentatie van de systemen en het bewijs van deze hierarchie wordt een beroep gedaan op de theorie van coalgebra's. Een coalgebraïsch resultaat dat aantoont dat zulke vertalingen verkregen kunnen worden uit injectieve natuurlijke transformaties wordt bewezen. Dit leidt tot een elegant bewijs van de hierarchie stelling. Voor zover bekend is dit de eerste toepassing van de theorie van coalgebra's op deze wijze.

Hiernaast richt dit proefschrift zich op een andere vorm van toestandsequivalentie, te weten zwakke bisimulariteit, welke gebaseerd is op de notie van zwakke bisimulatie. In de literatuur over concrete systemen is zwakke bisimulatie een bekend begrip. Ook voor probabilistische systemen bestaan verschil-

lende voorstellen voor zwakke bisimulatie. Echter, iets dergelijks ontbreekt in de theorie van coalgebra's. Een gedeeltelijke oplossing voor dit probleem voor actie-type coalgebra's wordt gepresenteers en wordt verantwoord door een correspondentieresultaat zowel voor standaard gelabelde transitiesystemen als voor één type probabilistische systemen te bewijzen.

Ten slotte worden een aantal onderwerpen verwant aan andere semantische relaties voor coalgebra's behandeld, te weten: simulatie, gekleurde transitiesemantiek, compositie van coalgebra's en manieren om paden in coalgebra's te definiëren. Probabilistische transitiesystemen worden hier gebruikt als leidende voorbeelden.

Curriculum Vitae

Ana Sokolova was born on 30th of May 1971 in Skopje, Macedonia.

She received her B.Sc. and her M.Sc. in computer science from Ss. Cyril and Methodius University in Skopje, Macedonia in 1994 and in 1999, respectively.

From 1995 until 2001 she worked as a teaching and research assistant within the Institute of Informatics, Faculty of Natural Sciences and Mathematics, at the Ss. Cyril and Methodius University in Skopje, Macedonia.

In June 2001 she became a Ph.D student at the Formal Methods Group, Department of Computer Science, Eindhoven University of Technology, The Netherlands.

From October 2005 she works as a postdoc researcher within the Security of Systems Group, Institute for Computing and Information Sciences, at the Radboud University Nijmegen.

Titles in the IPA Dissertation Series

J.O. Blanco. *The State Operator in Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1996-01

A.M. Geerling. *Transformational Development of Data-Parallel Algorithms*. Faculty of Mathematics and Computer Science, KUN. 1996-02

P.M. Achten. *Interactive Functional Programs: Models, Methods, and Implementation*. Faculty of Mathematics and Computer Science, KUN. 1996-03

M.G.A. Verhoeven. *Parallel Local Search*. Faculty of Mathematics and Computing Science, TUE. 1996-04

M.H.G.K. Kesseler. *The Implementation of Functional Languages on Parallel Machines with Distrib. Memory*. Faculty of Mathematics and Computer Science, KUN. 1996-05

D. Alstein. *Distributed Algorithms for Hard Real-Time Systems*. Faculty of Mathematics and Computing Science, TUE. 1996-06

J.H. Hoepman. *Communication, Synchronization, and Fault-Tolerance*. Faculty of Mathematics and Computer Science, UvA. 1996-07

H. Doornbos. *Reductivity Arguments and Program Construction*. Faculty of Mathematics and Computing Science, TUE. 1996-08

D. Turi. *Functorial Operational Semantics and its Denotational Dual*. Faculty of Mathematics and Computer Science, VUA. 1996-09

A.M.G. Peeters. *Single-Rail Handshake Circuits*. Faculty of Mathematics and Computing Science, TUE. 1996-10

N.W.A. Arends. *A Systems Engineering Specification Formalism*. Faculty of Mechanical Engineering, TUE. 1996-11

P. Severi de Santiago. *Normalisation in Lambda Calculus and its Relation to Type Inference*. Faculty of Mathematics and Computing Science, TUE. 1996-12

D.R. Dams. *Abstract Interpretation and Partition Refinement for Model Checking*. Faculty of Mathematics and Computing Science, TUE. 1996-13

M.M. Bonsangue. *Topological Dualities in Semantics*. Faculty of Mathematics and Computer Science, VUA. 1996-14

B.L.E. de Fluiter. *Algorithms for Graphs of Small Treewidth*. Faculty of Mathematics and Computer Science, UU. 1997-01

W.T.M. Kars. *Process-algebraic Transformations in Context*. Faculty of Computer Science, UT. 1997-02

P.F. Hoogendijk. *A Generic Theory of Data Types*. Faculty of Mathematics and Computing Science, TUE. 1997-03

T.D.L. Laan. *The Evolution of Type Theory in Logic and Mathematics*. Faculty of Mathematics and Computing Science, TUE. 1997-04

C.J. Bloo. *Preservation of Termination for Explicit Substitution*. Faculty of Mathematics and Computing Science, TUE. 1997-05

J.J. Vereijken. *Discrete-Time Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1997-06

F.A.M. van den Beuken. *A Functional Approach to Syntax and Typing*. Faculty of Mathematics and Informatics, KUN. 1997-07

A.W. Heerink. *Ins and Outs in Refusal Testing*. Faculty of Computer Science, UT. 1998-01

G. Naumoski and W. Alberts. *A Discrete-Event Simulator for Systems Engineering*. Faculty of Mechanical Engineering, TUE. 1998-02

J. Verriet. *Scheduling with Communication for Multiprocessor Computation*. Faculty of Mathematics and Computer Science, UU. 1998-03

J.S.H. van Gageldonk. *An Asynchronous Low-Power 80C51 Microcontroller*. Faculty of Mathematics and Computing Science, TUE. 1998-04

A.A. Basten. *In Terms of Nets: System Design with Petri Nets and Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1998-05

E. Voermans. *Inductive Datatypes with Laws and Subtyping – A Relational Model*. Faculty of Mathematics and Computing Science, TUE. 1999-01

H. ter Doest. *Towards Probabilistic Unification-based Parsing*. Faculty of Computer Science, UT. 1999-02

J.P.L. Segers. *Algorithms for the Simulation of Surface Processes*. Faculty of Mathematics and Computing Science, TUE. 1999-03

C.H.M. van Kemenade. *Recombinative Evolutionary Search*. Faculty of Mathematics and Natural Sciences, UL. 1999-04

E.I. Barakova. *Learning Reliability: a Study on Indecisiveness in Sample Selection*. Faculty of Mathematics and Natural Sciences, RUG. 1999-05

M.P. Bodlaender. *Scheduler Optimization in Real-Time Distributed Databases*. Faculty of Mathematics and Computing Science, TUE. 1999-06

M.A. Reniers. *Message Sequence Chart: Syntax and Semantics*. Faculty of Mathematics and Computing Science, TUE. 1999-07

J.P. Warners. *Nonlinear approaches to satisfiability problems*. Faculty of Mathematics and Computing Science, TUE. 1999-08

J.M.T. Romijn. *Analysing Industrial Protocols with Formal Methods*. Faculty of Computer Science, UT. 1999-09

P.R. D'Argenio. *Algebras and Automata for Timed and Stochastic Systems*. Faculty of Computer Science, UT. 1999-10

G. Fábián. *A Language and Simulator for Hybrid Systems*. Faculty of Mechanical Engineering, TUE. 1999-11

J. Zwanenburg. *Object-Oriented Concepts and Proof Rules*. Faculty of Mathematics and Computing Science, TUE. 1999-12

R.S. Venema. *Aspects of an Integrated Neural Prediction System*. Faculty of Mathematics and Natural Sciences, RUG. 1999-13

J. Saraiva. *A Purely Functional Implementation of Attribute Grammars*. Faculty of Mathematics and Computer Science, UU. 1999-14

R. Schiefer. *Viper, A Visualisation Tool for Parallel Program Construction*. Faculty of Mathematics and Computing Science, TUE. 1999-15

K.M.M. de Leeuw. *Cryptology and Statecraft in the Dutch Republic*. Faculty of Mathematics and Computer Science, UvA. 2000-01

T.E.J. Vos. *UNITY in Diversity. A stratified approach to the verification of distributed algorithms*. Faculty of Mathematics and Computer Science, UU. 2000-02

W. Mallon. *Theories and Tools for the Design of Delay-Insensitive Communicating Processes*. Faculty of Mathematics and Natural Sciences, RUG. 2000-03

W.O.D. Griffioen. *Studies in Computer Aided Verification of Protocols*. Faculty of Science, KUN. 2000-04

P.H.F.M. Verhoeven. *The Design of the MathSpad Editor*. Faculty of Mathematics and Computing Science, TUE. 2000-05

J. Fey. *Design of a Fruit Juice Blending and Packaging Plant*. Faculty of Mechanical Engineering, TUE. 2000-06

M. Franssen. *Cocktail: A Tool for Deriving Correct Programs*. Faculty of Mathematics and Computing Science, TUE. 2000-07

P.A. Olivier. *A Framework for Debugging Heterogeneous Applications*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2000-08

E. Saaman. *Another Formal Specification Language*. Faculty of Mathematics and Natural Sciences, RUG. 2000-10

M. Jelasity. *The Shape of Evolutionary Search Discovering and Representing Search Space Structure*. Faculty of Mathematics and Natural Sciences, UL. 2001-01

R. Ahn. *Agents, Objects and Events a computational approach to knowledge, observation and communication*. Faculty of Mathematics and Computing Science, TU/e. 2001-02

M. Huisman. *Reasoning about Java programs in higher order logic using PVS and Isabelle*. Faculty of Science, KUN. 2001-03

I.M.M.J. Reymen. *Improving Design Processes through Structured Reflection*. Faculty of Mathematics and Computing Science, TU/e. 2001-04

S.C.C. Blom. *Term Graph Rewriting: syntax and semantics*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2001-05

R. van Liere. *Studies in Interactive Visualization*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2001-06

A.G. Engels. *Languages for Analysis and Testing of Event Sequences*. Faculty of Mathematics and Computing Science, TU/e. 2001-07

J. Hage. *Structural Aspects of Switching Classes*. Faculty of Mathematics and Natural Sciences, UL. 2001-08

M.H. Lamers. *Neural Networks for Analysis of Data in Environmental Epidemiology: A Case-study into Acute Effects of Air Pollution Episodes*. Faculty of Mathematics and Natural Sciences, UL. 2001-09

T.C. Ruys. *Towards Effective Model Checking*. Faculty of Computer Science, UT. 2001-10

D. Chkliaev. *Mechanical verification of concurrency control and recovery protocols*. Faculty of Mathematics and Computing Science, TU/e. 2001-11

M.D. Oostdijk. *Generation and presentation of formal mathematical documents*. Faculty of Mathematics and Computing Science, TU/e. 2001-12

A.T. Hofkamp. *Reactive machine control: A simulation approach using χ*. Faculty of Mechanical Engineering, TU/e. 2001-13

D. Bošnački. *Enhancing state space reduction techniques for model checking*. Faculty of Mathematics and Computing Science, TU/e. 2001-14

M.C. van Wezel. *Neural Networks for Intelligent Data Analysis: theoretical and experimental aspects*. Faculty of Mathematics and Natural Sciences, UL. 2002-01

V. Bos and J.J.T. Kleijn. *Formal Specification and Analysis of Industrial Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2002-02

T. Kuipers. *Techniques for Understanding Legacy Software Systems*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2002-03

S.P. Luttik. *Choice Quantification in Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-04

R.J. Willemen. *School Timetable Construction: Algorithms and Complexity*. Faculty of Mathematics and Computer Science, TU/e. 2002-05

M.I.A. Stoelinga. *Alea Jacta Est: Verification of Probabilistic, Real-time and Parametric Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-06

N. van Vugt. *Models of Molecular Computing*. Faculty of Mathematics and Natural Sciences, UL. 2002-07

A. Fehnker. *Citius, Vilius, Melius: Guiding and Cost-Optimality in Model Checking of Timed and Hybrid Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-08

R. van Stee. *On-line Scheduling and Bin Packing*. Faculty of Mathematics and Natural Sciences, UL. 2002-09

D. Tauritz. *Adaptive Information Filtering: Concepts and Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2002-10

M.B. van der Zwaag. *Models and Logics for Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-11

J.I. den Hartog. *Probabilistic Extensions of Semantical Models*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2002-12

L. Moonen. *Exploring Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-13

J.I. van Hemert. *Applying Evolutionary Computation to Constraint Satisfaction and Data Mining*. Faculty of Mathematics and Natural Sciences, UL. 2002-14

S. Andova. *Probabilistic Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2002-15

Y.S. Usenko. *Linearization in μCRL*. Faculty of Mathematics and Computer Science, TU/e. 2002-16

J.J.D. Aerts. *Random Redundant Storage for Video on Demand*. Faculty of Mathematics and Computer Science, TU/e. 2003-01

M. de Jonge. *To Reuse or To Be Reused: Techniques for component composition and construction*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-02

J.M.W. Visser. *Generic Traversal over Typed Source Code Representations*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-03

S.M. Bohte. *Spiking Neural Networks*. Faculty of Mathematics and Natural Sciences, UL. 2003-04

T.A.C. Willemse. *Semantics and Verification in Process Algebras with Data and Timing*. Faculty of Mathematics and Computer Science, TU/e. 2003-05

S.V. Nedea. *Analysis and Simulations of Catalytic Reactions*. Faculty of Mathematics and Computer Science, TU/e. 2003-06

M.E.M. Lijding. *Real-time Scheduling of Tertiary Storage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-07

H.P. Benz. *Casual Multimedia Process Annotation – CoMPAs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-08

D. Distefano. *On Modelchecking the Dynamics of Object-based Software: a Foundational Approach*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-09

M.H. ter Beek. *Team Automata – A Formal Approach to the Modeling of Collaboration Between System Components*. Faculty of Mathematics and Natural Sciences, UL. 2003-10

D.J.P. Leijen. *The λ Abroad – A Functional Approach to Software Components*. Faculty of Mathematics and Computer Science, UU. 2003-11

W.P.A.J. Michiels. *Performance Ratios for the Differencing Method*. Faculty of Mathematics and Computer Science, TU/e. 2004-01

G.I. Jojgov. *Incomplete Proofs and Terms and Their Use in Interactive Theorem Proving*. Faculty of Mathematics and Computer Science, TU/e. 2004-02

P. Frisco. *Theory of Molecular Computing – Splicing and Membrane systems*. Faculty of Mathematics and Natural Sciences, UL. 2004-03

S. Maneth. *Models of Tree Translation*. Faculty of Mathematics and Natural Sciences, UL. 2004-04

Y. Qian. *Data Synchronization and Browsing for Home Environments*. Faculty of

Mathematics and Computer Science and Faculty of Industrial Design, TU/e. 2004-05

F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-06

L. Cruz-Filipe. *Constructive Real Analysis: a Type-Theoretical Formalization and Applications*. Faculty of Science, Mathematics and Computer Science, KUN. 2004-07

E.H. Gerding. *Autonomous Agents in Bargaining Games: An Evolutionary Investigation of Fundamentals, Strategies, and Business Applications*. Faculty of Technology Management, TU/e. 2004-08

N. Goga. *Control and Selection Techniques for the Automated Testing of Reactive Systems*. Faculty of Mathematics and Computer Science, TU/e. 2004-09

M. Niqui. *Formalising Exact Arithmetic: Representations, Algorithms and Proofs*. Faculty of Science, Mathematics and Computer Science, RU. 2004-10

A. Löh. *Exploring Generic Haskell*. Faculty of Mathematics and Computer Science, UU. 2004-11

I.C.M. Flinsenberg. *Route Planning Algorithms for Car Navigation*. Faculty of Mathematics and Computer Science, TU/e. 2004-12

R.J. Bril. *Real-time Scheduling for Media Processing Using Conditionally Guaranteed Budgets*. Faculty of Mathematics and Computer Science, TU/e. 2004-13

J. Pang. *Formal Verification of Distributed Systems*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-14

F. Alkemade. *Evolutionary Agent-Based Economics*. Faculty of Technology Management, TU/e. 2004-15

E.O. Dijk. *Indoor Ultrasonic Position Estimation Using a Single Base Station*. Faculty of Mathematics and Computer Science, TU/e. 2004-16

S.M. Orzan. *On Distributed Verification and Verified Distribution*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-17

M.M. Schrage. *Proxima - A Presentation-oriented Editor for Structured Documents*. Faculty of Mathematics and Computer Science, UU. 2004-18

E. Eskenazi and A. Fyukov. *Quantitative Prediction of Quality Attributes for Component-Based Software Architectures*. Faculty of Mathematics and Computer Science, TU/e. 2004-19

P.J.L. Cuijpers. *Hybrid Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2004-20

N.J.M. van den Nieuwelaar. *Supervisory Machine Control by Predictive-Reactive Scheduling*. Faculty of Mechanical Engineering, TU/e. 2004-21

E. Ábrahám. *An Assertional Proof System for Multithreaded Java -Theory and Tool Support-* . Faculty of Mathematics and Natural Sciences, UL. 2005-01

R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02

C.N. Chong. *Experiments in Rights Control - Expression and Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-03

H. Gao. *Design and Verification of Lock-free Parallel Algorithms*. Faculty of Mathematics and Computing Sciences, RUG. 2005-04

H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05

M.T. Ionita. *Scenario-Based System Architecting - A Systematic Approach to Developing Future-Proof System Architectures*. Faculty of Mathematics and Computing Sciences, TU/e. 2005-06

G. Lenzini. *Integration of Analysis Techniques in Security and Fault-Tolerance*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-07

I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, UU. 2005-09

O. Tveretina. *Decision Procedures for Equality Logic with Uninterpreted Functions*. Faculty of Mathematics and Computer Science, TU/e. 2005-10

A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14

M.R. Mousavi. *Structuring Structural Operational Semantics*. Faculty of Mathematics and Computer Science, TU/e. 2005-15

A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. Faculty of Mathematics and Computer Science, TU/e. 2005-16