View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Catalogo dei prodotti della ricerca

Federica Panarotto

# Measures on probabilistic automata

May 5, 2017

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
Prof. Roberto Segala

Series N°: **TD**

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

*To Giovanni and Anna,*
*those who taught me to be curious and tenacious.*

*A Giovanni e Anna,*
*coloro che mi hanno insegnato ad essere curiosa e tenace.*

La fatica è una realtà inevitabile, mentre la possibilità di farcela o meno è a esclusiva discrezione di ogni individuo.

H. Murakami- L'arte di correre

# Ringraziamenti

Un ringraziamento speciale a prof. Roberto Segala che ha saputo insegnarmi l'essenza di fare ricerca, specialmente la capacit di saper scovare gli errori che si possono fare in questo lavoro, dai minuscoli ai macroscopici. Un ringraziamento a prof. Paolo Fiorini per avermi saputo guidare, assieme al prof. Roberto, verso la fine del dottorato. Un ringraziamento a prof. Mario Bravetti e prof. Francesco Ranzato per avermi aiutato a migliorare questa tesi. Un ringraziamento a prof. Jiannis Pachos e Giandomenico Palumbo per avermi aiutata a capire la fisica quantistica. Ringrazio Peter Selinger per essere stato preciso nella correzione della prima tesi e, soprattutto, lavorativamente corretto.

Ringrazio i dottorandi e i post-doc delle stanze 1.64/A e 1.71 del Dipartimento di informatica con cui ho condiviso pause pranzo e discussioni pi o meno incentrate sulla ricerca e che mi hanno regalato mille cartoline. Un particolare ringraziamento ai dottori di ricerca Gabriele Pozzani, Francesca Pizzorni, Vincenzo Bonnici, Alberto Sabaini, Pietro Sala per aver condiviso gioie e patimenti in questo lungo mio dottorato, in particolare durante il periodo pi difficile del dottorato. Ringrazio Chiara De Rosa e Alessia Maiore per avermi aiutata e sopportato durante gli anni in cui abbiamo condiviso la casa finch ero una stressata dottoranda.

Un particolare ringraziamento a tutti quelli che mi hanno regalato una cartolina da tutti i luoghi dove c'era una conferenza o un workshop, dove hanno vissuto per qualche mese, dove semplicemente hanno fatto una vacanza.

Ringrazio la mia famiglia, Anna Fratta, Giovanni e Silvia Panarotto, Nicola Dal Dosso, Alessandro e Roberta Pernigotto, Giulia, Giovanni e Ornella Fratta, per il fondamentale supporto.

Ultima, ma solo per una quesitone temporale, ringrazio la mia squadra di podismo, l'Atletica lupatotina, con cui ho corso e consumato lo stress da dottorato lungo i molti kilometri di allenamento assieme e le frequenti mezze maratone portate a termine via via con i vari componenti della squadra.

# Abstract

In this thesis we consider nondeterministic probabilistic processes modeled by automata. Our purpose is the analysis of the problem of approximated bisimulations. These relations are used, generally, to simplify the models of some systems and to model agents and attackers in security protocols. For the latter field there are several proposals to use metrics, which are the quantitative analogue of probabilistic bisimilarity and allow a greater precision. A metric is about a degree of similarity between states. Starting from the formalisation of approximate (bi)simulation given in [62], we define two metrics on states and on distributions. These metrics are based on the concept of error allowed during the simulation of a state with respect to another one. We investigate the relation between these metrics with a largely used one, the Kantorovich metric, and discover that they are equivalent. Then we recast for probabilistic automata the transformer of measures proposed by De Alfaro et al., obtaining a new functional $F$ that is a conservative extension of the transformers proposed in the literature. We show that the minimum fix point of $F$ coincides with its over-aproximated by the measures derived from [62], thus showing the existence of a strict relation between the Turrinis approximate bisimulations with the literature on metrics.

# Sommario

In questa tesi consideriamo i processi probabilistici non-deterministici modellati attraverso automi. Il nostro obiettivo è l'analisi dei problemi di bisimulazioni approssimate. Queste relazioni sono usate, generalmente, per semplificare i modelli di alcuni sistemi e per modellare agenti e attaccanti nei protocolli di sicurezza. In questo ultimo campo ci sono diversi proposte di utilizzo di metriche, le quali sono l'analogo quantitativo della bisimulazione probabilistica e permettono una miglior precisione. Una metrica è grossomodo un grado di similarità tra stati. Iniziando dalla formalizzazione di (bi)simulazione approssimata data in [62], definiamo due metriche su stati e su distribuzioni. Queste metriche sono basate sul concetto di errore ammesso durante la simulazione di uno stato rispetto un altro stato. Investigheremo la relazione tra queste metriche con una metrica largamente utilizzata, la metrica di Kantorovich, e scopriremo che esse sono equivalenti. Poi riadatteremo per gli automi probabilistici il trasformatore di misure proposto da De Alfaro e al., ottenendo un nuovo funzionale $F$ che è una estensione conservativa dei trasformatori proposti in letteratura. Mostreremo che il minimo punto fisso di $F$ coincide con la sua sovra-approssimazione dalle misure derivate da [62], attraverso la dimostrazione dell'esistenza di una stretta relazione tra le bisimulazioni approssimate di Turrini con le metriche in letteratura.

# Contents

**1 Introduction 1**

**2 Related work 7**

**3 Background 11**

3.1 Measures . . . . . . . . . . 11

3.1.1 $\sigma$-algebra . . . . . . . . . . 11

3.1.2 Borel $\sigma$-algebra . . . . . . . . . . 12

3.2 Probabilistic systems . . . . . . . . . . 14

3.2.1 General processes . . . . . . . . . . 14

3.2.2 Probabilistic automata . . . . . . . . . . 17

3.3 Bisimulation . . . . . . . . . . 18

3.3.1 Classical bisimulation . . . . . . . . . . 20

3.3.2 Probabilistic bisimulation . . . . . . . . . . 23

3.3.3 Bisimilarity as a fixed-point . . . . . . . . . . 29

3.4 Metric theory . . . . . . . . . . 33

3.4.1 Metric instead of logic . . . . . . . . . . 33

3.4.2 Kantorovich metric . . . . . . . . . . 35

3.4.3 Metric and bisimulation . . . . . . . . . . 37

**4 Security with $\varepsilon$-sim. 41**

4.1 Security and cryptography . . . . . . . . . . 41

4.2 Polynomially accurate simulation relation . . . . . . . . . . 43

4.3 $\varepsilon$-(bi)simulation . . . . . . . . . . 46

4.3.1 $\varepsilon$-lifting . . . . . . . . . . 47

4.3.2 $\varepsilon$-(bi)simulation . . . . . . . . . . 49

**5 From $\varepsilon$-sim.s to metrics 55**

5.1 Transitivity of $\varepsilon$-lifting . . . . . . . . . . 55

5.2 $\varepsilon$-relation . . . . . . . . . . 58

5.3 Probabilistic metrics with $\varepsilon$-lifting . . . . . . . . . . 60

5.4 Equivalence of pseudometrics . . . . . . . . . . 63

5.5 Bisimilarity as fixed-point of the operator F . . . . . . . . . . 65

**6 Conclusions 71**

**References** **75**

# List of Diagrams

D.3.1 Classical simulation . . . 20
D.3.2 Two nondeterministic transitions . . . 20
D.3.3 Classical simulation of a nondeterministic transition . . . 21
D.3.4 Classical bisimulation . . . 21
D.3.5 Classical bisimilarity . . . 21
D.3.6 Classical weak simulation . . . 22
D.3.7 Classical weak bisimulation . . . 22
D.3.8 Two probabilistic transitions . . . 23
D.3.9 Probabilistic simulation . . . 27
D.3.10 Probabilistic bisimulation with double diagram . . . 28
D.3.11 Probabilistic bisimulation . . . 28
D.3.12 Probabilistic bisimilarity . . . 28
D.4.1 $\varepsilon$-lifting relation . . . 47
D.4.2 Lifting relation between two states . . . 48
D.4.3 $\varepsilon$-lifting relation between two states . . . 50
D.4.4 $\varepsilon$-simulation, simplify diagram . . . 50
D.4.5 $\varepsilon$ equivalence, simplify diagram . . . 51
D.4.6 $\varepsilon$-bisimulation, simplify diagram . . . 51
D.4.7 $\varepsilon$-bisimilarity, simplify diagram . . . 52
D.4.8 $\varepsilon$-lifting relation between two states with zero error . . . 52
D.4.9 Probabilistic transitions with maximum error . . . 53
D.5.1 Transitivity of probabilistic transitions . . . 56
D.5.2 Transitivity of $\varepsilon$-lifting, simplify diagram . . . 56
D.5.3 $\varepsilon$-relation, simplify diagram . . . 58
D.5.4 Transitivity of $\varepsilon$-relation bisimilarity, simplify diagram . . . 59
D.5.5 Distance $d_{L,R}$ on distributions . . . 61
D.5.6 Distance $d_{L,R}$ on states . . . 62
D.5.7 Iterator operator F . . . 67
D.5.8 Lattice of distances . . . 67
D.5.9 Monotony of iterator F . . . 68
D.6.1 Iterator F cfr. iterator H . . . 72

# 1

# Introduction

The evolution of communication networks has led toward increasingly complex communication protocols to interconnect heterogeneous systems. To function properly these protocols require formal methodologies for verification, implementation and testing [8]. The development of a formal specification provides several advantages:

- insights and an understanding of the software requirements and software design;
- reveal and remove ambiguity, inconsistency and incompleteness;
- facilitate communication of requirement or design;
- provides a basis for an elegant software design;
- traceability.

The use of formal methods for software and hardware design and for verification of the requirements performs an appropriate mathematical analysis, that contributes to the reliability and robustness of a design and to verify the completely and accurately implementation. A formal specification is the notion at the heart of formal methods. Once a formal specification has been produced, it is a guide while the concrete system is developed during the design process. After the development, the specification may be used as the basis for proving properties and by inference the developed system.

We use formal methods in several fields of computer science, from the managing of the complexity of large systems to the reasoning about distributed systems, i.e. about concurrency, passing through the formal verification of security protocols. We consider the computer security, in several networks the attackers have access to every message and can perform a statistical analysis of intercepted messages to obtain information. To preserve secrecy or anonymity we model agents, including attackers, as processes in some formal system and use tools, like model checkers, to verify properties of the protocol [39, 48]. In this way we check if an agents can be substitute with an attacker by introducing a concept of equivalence between processes. We consider morphisms that are "structure-preserving" maps. The most basic forms of morphism are the homomorphisms that essentially give us a way of embedding a structure, the source, into another one, the target, such that all the relations in the source are present in the target. The converse need not be true, for this reason stronger notions of morphism are needed. One such notion is isomorphism, where strong-isomorphic structures must be the same, i.e., "algebraically identical". The operations in processes are partial operations, thus we generalise the notion of isomorphism with a notion in between homomorphism and isomorphism, i.e., weak homomorphism. The extension from the algebraic notion of weak homomorphism into the study of sequential, imperative, and non-terminating programs done by Milner [42] is called bisimulation. This name is a term that better conveys the imitation of an operation of a system by another system. Since this is a fundamental notion in the thesis, we show an example of bisimulation. For instance, isomorphic graphs have the same number of nodes, which need not be the case for bisimilar graphs. In rooted directed graphs a bisimulation is coarser than graph isomorphism because, intuitively, bisimulation allows us to observe a graph only through the movements that are possible along its edges. By contrast, with isomorphisms the identity of the nodes is observable too. The idea of bisimulation is interesting in two ways: to abstract some irrelevant detail from programs to come closer to a definition of algorithm, and as a manageable technique for proving simulation between programs, which in some cases may make easier the task of proving a program correct.

Probabilistic model allows one to analyse situations where the attacker uses statistical techniques to extract information, more in general model quantitative processes. We model these given quantity by probability, as the probability that the step will happen [16,37,58] or the resources needed to perform that step [11,47,68]. Its use is motivated also by the fact that there are many instances of real-life systems whose behaviour can be accurately modelled by considering their stochastic characteristics. In fact some real-life systems can be inherently stochastic in nature, because they include components which are known to be unreliable or because the exact timing of inputs to system remain unpredictable, e.g., computer networks. Most existing process calculi for security lack of these probabilistic constructs. To develop a formalism to represent probabilistic security protocols and systems the calculus have to incorporate mechanisms to express probabilistic choice as done by [61, 62]. We consider a single-valued and total transition function $Tr$ which represents a sequence of operation in a program. The development of probabilistic model considers computation trees rather than sequences, i.e. we consider simulation of parallel programs. There exists a noticeable difference between the development of simulation and the operationally definition of weak homomorphism. This difference is caused by the nature of the processes studied: in weak homomorphism the processes considered are deterministic, thus the alternation of universal and existential quantifiers does not play a role. The standard notion of bisimulation can be adapted to these probabilistic systems by treating the probabilistic quantities as labels, for example [14, 16, 37, 52, 58]. Probabilistic formal methods [32, 37, 59], that describe the transitions by probabilistic distributions, allow to generalise the concept of bisimulations to the field of probabilistic systems with the notions of approximate (bi)simulations [62]. The latter differs from the other probabilistic systems since a transition from a state $s$ leads to a distribution $\mu$ on just a single successor state, instead of leading to a successor distribution over states. A successor distribution of $\mu$ gives the probability $\mu(s')$ of entering successor state $s'$. This probabilistic transition structure is reflected in the definition of simulation. A binary relation $R$ is a simulation relation if, for all $(s, t) \in R$, $t$ can mimic all stepwise behaviour of $s$ with respect to $R$. Intuitively, this means that every distribution $\mu$ leaving state $s$ with label $a$ has a distribution $\mu'$ leaving state $t$ with the same label $a$ such that the distributions $\mu$ and $\mu'$ are related: relations between distributions are established by weight functions [32]. This is the notion of bisimulation, that lacks robustness as against system parameter perturbations. Thus it is often useful to identify similar objects that differ only for a small value, i.e. for an error. These approximate notions appear much less restrictive than the exact one, which does not provide a robust relation, since quantities are matched only when they are identical. For instance, processes that differ for a very small probability would be considered just as different as processes, that perform completely different actions. This is particularly relevant to security systems where specifications can be given as perfect, but impractical processes and other, practical processes are considered safe if they only differ from the specification with a negligible probability.

A second development of the notion of bisimulation [43] is a proof technique for giving an inductive definition, which tells us what are the constructors for generating all the elements. The inductive proof principle allows us to infer the inductive set as a set T closed forward, i.e., for each rule whose premise is satisfied in T there is an element of T such that the element is the conclusion of the rule. Since the definition is inductive, then $T$ is the smallest universe in which such rules live. We model the rule as a monotone operator, then $T$ is a least fixed point of this operator that yields an inductive definition for trying to define a bisimulation. In a consecutive refinement [46], with Park's contribution [49], the inductive definition turns into a coinductive one. A coinductive definition tells us what are the destructors that decompose the elements. Contrary to the constructors, the destructors show what we can observe of the elements. The coinductive proof principle allows us to infer that a set $T$ is included in the coinductive set by proving that the given set satisfies the backward closure, i.e., for each element of T there is a rule whose premise is satisfied in T such that the element is the conclusion of the rule. With respect to the previous proof the coinduction is the mathematical dual to structural induction. As above, we model the rule as a monotone operator, then $T$ is a greatest fixed point of this operator that yields an coinductive definition for defining a bisimulation. Induction and coinduction is a technique for defining and proving behavioural properties of systems of concurrent interacting objects studied in [2,45,54,55]. These definitions are based on recursive function theory and fixed point iteration [6,49,56].

The standard notion of bisimulation for probabilistic or stochastic systems cannot distinguish between two processes that are substantially different and two processes that differ by only a small amount in a real valued parameter. It is often more useful to say how similar two processes are than to say whether they are exactly the same. This idea leads to the development of metrics, which are robust to small perturbations of the model parameters. For instance, we consider the case where we want to know whether two processes are behaving in a similar way, of sure they differ by only a small amount in real-valued parameters. Thus we define a metric on the set of processes of the minimal process algebra that will measure how much two processes are alike in terms of behaviour. In this sense, two processes are at distance zero if and only if they are bisimilar. Thus, the metrics will be quantitative extensions of the notion of bisimulation. The metrics assign a real number in the interval $[0, 1]$, i.e. a distance, to each pair of states of the probabilistic transition system. The distance captures the behavioural similarity of the states, smaller is the distance more the states behave alike. The metrics between processes are a quantitative analogue of probabilistic bisimilarity and can be introduced essentially in two separate ways. The first approach employs the probabilistic conditional kernels underlying the stochastic processes under study - in this sense, the approximation comes from metrics between (marginals of) probability measures related to the two processes. The second procedure looks at distance metrics between trajectories of the two processes and utilizes the dynamical properties of the two processes to define such metrics: this can be done either by analyzing the models syntax, or by directly employing their semantics in order to compare realizations of the two models.

The approximate bisimulation relations and the metrics are two approaches for analysing the errors existing between two similar probabilistic systems, which describe processes. As we have discussed above, these approximations are very useful in security protocols. But the connection between these two approaches is, at the moment, unknown. In this thesis we formalise this problem and show that the approximate approach followed by Segala and Turrini [62] is closely related to the metric of Van Breugel [63,65] and of Desharnais [17,18]. In the following scheme we give a detailed description of the contributions of the thesis.

1. Starting from the definition of approximate (bi)simulation [61,62] it is possible defining a pseudo-metric on states and probabilistic measures. The distance between two states is the infimum of the errors that can be done. The thesis shows that this is an infimum of the pseudo metric.
2. The infimum of the error whereby two measures can be put in relation, starting from a relation R on states, is the Kantorovich distance between two measures starting by a pseudo-metric $d_R$ which is 0 on pairs of states in relation and 1 elsewhere.

3. We can define a metric transformer $H$ [65] on probabilistic automata compatible with the functional transformers defined in literature. Consequently the metrics can also be studied on probabilistic automata. If on the automata we impose restrictions that have other models, then the transformer defined in the thesis coincides with the literature transformers.
4. The pseudo-metric of point 1 is the result of the transformer H of point 3 applied to the pseudo-metric $d_R$ of point 2. Furthermore $H(d_R) \leqslant d_R$. Consequently the minimum fixed point of H, i.e. $d_R$ is an over approximation of the metric on probabilistic automata consistent with the literature. From here the result that the approximate simulations are a sound method of demonstrating upper limits to the distance between two automata.

## Overview of the Thesis

The thesis is structured as follows.

Chapter 1. We give an overview of the issues discuss in the thesis, especially we give a historical perspective on the bisimulation relation and its refinements generated by considering the probabilistic systems. The latter are the approximate bisimulations and the metrics, which are useful above all in security field. Nevertheless in security literature there is a hole in the study of the metrics in this field.

Chapter 2. In literature we can find several papers that are focused on probabilistic bisimulation based on several model for describing probabilistic systems and on metrics between probabilistic transitions. In this chapter we show the related works existing.

Chapter 3. We introduce some preliminary notions about mathematical, probabilistic systems, bisimulation relations, metric theory. We characterise our definitions with an operational overview and we do not refer to logic, these two characterisations are related in the paragraph Metric instead of logic. In this background we analyse the definition bisimulation as fixed point and the definition of Kantorovich distance, and De Alfaro et al. operator.

Chapter 4. There exist few papers where the approximate bisimulations are used to to field of security and cryptography, we show here the study and the result obtained. These analysis are at the basis of our work and may be a guideline for future works.

Chapter 5. Based on the concept of lifting with, we introduce the approximate (bi)simulation with error and relation with error and show some properties and introduce. The main notions of this chapter are probabilistic metric defined as the small error that verifies the previous relations. This metric is closely related, precisely equivalent, with Kantorovich metric. We define a transformer iterator, as De Alfaro et al. operator, whose fixed point is our metric. This operator is the basic step for generating a upper limits to the distances between two automata and use these metric in security and cryptography field.

Chapter 6. The last chapter contain the conclusions of the thesis, principally draw attention to the relations between our metric and operator and the metric and operator present in literature. In the last part we have inserted the future works, one related to the interesting field of security and another related to the definition of a limit for our metric.

## 2

# Related work

In this chapter we give an extensive overview of existing work on bisimulation, approximate bisimulation, and metrics.

Concurrent computing is a programming paradigm based on a form of modular programming, i.e. it factors an overall computation into subcomputations that may be executed concurrently. Pioneers in the field of concurrent computing include Edsger Dijkstra [20], Per Brinch Hansen, and C.A.R. Hoare [29]. In the years a wide variety of formalisms for modelling and understanding concurrent systems have been developed, among which the model of Petri nets [51] and Hennessy-Milner logic [44]. The basic description of a computer system is as a state machine that computes by moving from one state to another state. This leads to the idea of Labelled Transition Systems (LTS) [34, 53]. These systems have been used successfully for the modelling of ordinary distributed systems [23, 31, 40, 41, 45], and for their verification [67].

The concept of strong probabilistic bisimulation over a discrete-time, finite-state Markov chain has been introduced in [37], based on earlier notions for non-probabilistic models [44, 50]. The work in [26] uses similar notions for Markov decision processes with finite state spaces, and puts forward procedures for finding factored bisimilar models. The notion of weak bisimulation is discussed in [4,28,52] for a number of (finite-state) probabilistic processes. The contributions in [32, 60] cover the notion of probabilistic simulation relations for classes of probabilistic automata. [5] provide a recapitulation and draw relationships between these notions. The interesting work in [19] discusses approximate notions of bisimulations for finite state labelled Markov chains, and elaborates on this notions by using a logical approach as well as one based on games. The use of approximate notions is advocated in [24] and motivated by robustness issues related to the verification of specifications over probabilistic models. Furthermore, approximate notions appear much less restrictive than the exact one, particularly when applied over models with continuous state spaces - this is precisely what has been observed also for deterministic models, where notions of exact bisimulation have been developed only for limited classes of models. The introduction of approximate versions [25] based on distance between trajectories of deterministic models has lead to the study of approximate abstractions for nonlinear and switched systems. Building on these results, the material in [17] is relevant in that metrics for labelled Markov processes are discussed, whereas [18] proposes metrics via weak bisimulations, and the contributions in [21, 22] discusses metrics for respectively finite- and infinite-state Markov decision processes. For more details see [1].

Giacalone et al. [24] were the first to suggest a metric between probabilistic transition systems to formalize the notion of distance between processes. Metrics were used also in [36] to give denotational semantics for reactive models. De Vink and Rutten [13] showed that discrete probabilistic transition systems can be viewed as coalgebras [17, 18, 64, 65]. Desharnais et al. [17] studied a logical pseudometric for labelled Markov chains, which is a reactive model of probabilistic systems. The metric has the property that two processes have distance of 0 if and only if they are probabilistic bisimilar. They also introduced a probabilistic process calculus and showed that some of the process constructors are non-expansive. A similar pseudometric was defined by van Breugel and Worrell [65] via the terminal coalgebra of a functor based on a metric on the space of Borel probability measures. Interestingly, van Breugel and Worrell [64] also presented a polynomial-time algorithm to approximate their coalgebraic distances. In [18] Desharnais et al. dealt with labelled concurrent Markov chains (this model can be captured by the simple probabilistic automata of [57]). They showed that the greatest fixed point of a monotonous function on pseudometrics corresponds to the weak probabilistic bisimilarity of [52]. They also showed that some process constructors of a probabilistic process calculus are non-expansive.

The first proposal based on metrics was by Giacalone et al. [24] for deterministic probabilistic processes. Later, Desharnais et al. [17, 18] and van Breugel and Worrell [64, 65] investigated the notion of metric for more general probabilistic systems, using much more sophisticated techniques to deal with the combination of probabilistic distribution, nondeterminism and recursion. In particular, they used the notion of Hutchinson metric [30] on distributions; this metric is also known under many different names including Kantorovich metric [33] and Vaserstein metric [66]. In [17, 18], Desharnais et al. treated the case of labelled Markov chains and labelled concurrent Markov chains respectively, and defined the intended metric as the greatest fixed point of a monotonous function. In contrast, the authors of [64, 65] used a construction based on the (unique) fixed point of a contractive transformation. They considered similar classes of automata, namely fully probabilistic systems and reactive models.

We model these given quantity by probability, as the probability that the step will happen [16, 37, 58] or the resources needed to perform that step [11, 47, 68]. Thus the standard notion of bisimulation can be adapted to these probabilistic systems by treating the probabilistic quantities as labels, for example [16, 37, 52, 58]. Probabilistic formal methods [32, 37, 59], that describe the transitions by probabilistic distributions, allow to generalise the concept of bisimulations to the field of probabilistic systems with the notions of approximate (bi)simulations [62].

# 3

# Background

Along the chapter we recall some basic notions and properties we will use in the rest of the thesis. In Section 3.1 we introduce the fundamental arguments of Measure theory. In Section 3.2 we introduce probabilistic systems and, in particular, automata. In Section 3.3 we introduce bisimulation relations. In Section 3.4 we introduce Metric theory.

## 3.1 Measures

Measure theory is the study of measures, it generalises the intuitive notions of length, area, and volume. In this section we give the basic definitions of measure theory, we recall the fundamental definition of $\sigma$-algebra and Borel $\sigma$-algebra. More information can be found in [10, 35].

### 3.1.1 $\sigma$-algebra

A $\sigma$-algebra on a set $\Omega$ is a collection of subsets of a set $\Omega$ that contains $\emptyset$ and $\Omega$, and is closed under complements, finite unions, countable unions, and countable intersections.

**Definition 3.1 ($\sigma$-algebra).** *We say that $F \subseteq 2^{\Omega}$ is a $\sigma$-algebra (or a $\sigma$-field), if*

*a) $\Omega \in F$ and $\emptyset \in F$*
*b) if $A \in F$ then $A^c \in F$, where $A^c = \Omega \setminus A$*
*c) if $A_i \in F$ for $i = 1, 2, \dots$ then $\bigcup_i A_i \in F$*
*c') if $A_i \in F$ for $i = 1, 2, \dots$ then $\bigcap_i A_i \in F$.*

The conditions c) and c') are equivalent for De Morgans law, which is $(\bigcup_i A_i^c)^c = \bigcap_i A_i$. If $S$ is any collection of subsets of $F$, then we can always find a $\sigma$-algebra containing $S$, namely the power set of $F$. By taking the intersection of all $\sigma$-algebras containing $S$, we obtain the smallest such $\sigma$-algebra.

**Definition 3.2 ($\sigma$-algebra generators).** *The $\sigma$-algebra* generated *by $S$ is the smallest $\sigma$-algebra containing $S$.*

Consider a set $\Omega$. A $\sigma$-algebra on $\Omega$ is a set $F \subseteq 2^{\Omega}$ that includes $\Omega$ and is closed under complement and countable union. A *measurable space* is pair $(\Omega, F)$ where $\Omega$ is a set, also called *sample space*, and $F$ is a $\sigma$-field over $\Omega$. A measurable space $(\Omega, F)$ is called discrete if $F = 2^{\Omega}$.

A measure is a countably additive, non-negative, extended real-valued function defined on a $\sigma$-algebra.

**Definition 3.3 (Measure).** *For each countable collection $\{\Omega_i\}_{i \in I}$ of pairwise disjoint elements of $F$, a* measure *over a measurable space $(\Omega, F)$ is a function $\mu \colon F \to R^+ > 0$ such that $\mu(\bigcup_I \Omega_i) = \sum_I \mu(\Omega_i)$.*
*A* probability measure *over a measurable space $(\Omega, F)$ is a measure $\mu$ over $(\Omega, F)$ such that $\mu(\Omega) = 1$.*

**Definition 3.4 (Discrete measure).** *A* discrete measure *is a measure over a discrete measurable space $(\Omega, 2^{\Omega})$. We denote by $Disc(\Omega)$ the set of discrete probability measures over the set $\Omega$.*

We call a discrete probability measure a *Dirac measure*, denoted by $\delta_x$, if it assigns measure 1 to exactly one object $x$; in a natural way we define $\delta_x$ by

$$\delta_x(\Omega) = \begin{cases} 1 & \text{if } x \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

A *sub-probability measure* over $(\Omega, F)$, is a measure over $(\Omega, F)$ such that $\mu(\Omega) \leqslant 1$. The set of discrete sub-probability measures over the set $\Omega$ is denoted by $\mathrm{SubDisc}(\Omega)$. The *support* of a measure $\mu$ over $(\Omega, F)$, denoted by $\mathrm{Supp}(\mu)$, is the set $\{\, \omega \in \Omega \mid \mu(\omega) > 0 \,\}$. A *probability space* is a triple $(\Omega, F, \rho)$, where $(\Omega, F)$ is a measurable space and $\rho$ is a probability measure on $(\Omega, F)$. Let $(\Omega_1, F_1)$ and $(\Omega_2, F_2)$ be two measurable spaces. A function $f \colon \Omega_1 \to \Omega_2$ is said to be a *measurable function* from $(\Omega_1, F_1)$ to $(\Omega_2, F_2)$ if the inverse image under f of any element of $F_2$ is an element of $F_1$. In this case, given a measure $\rho$ on $(\Omega_1, F_1)$ it is possible to define a measure on $(\Omega_2, F_2)$ via $f$, called the *image measure* of $\rho$ under $f$ and denoted by $f(\rho)$, as follows. For each $X \in F_2, f(\rho)(X) = \rho(f^{-1}(X))$. In other words, the measure of $X$ in $F_2$ is the measure in $F_1$ of those elements whose $f$-image is in $X$. The measurability of f ensures that $f(\rho)$ is indeed a well defined measure.

### 3.1.2 Borel $\sigma$-algebra

A $\sigma$-algebra which is related to the topology of a set is the Borel $\sigma$-algebra.

**Definition 3.5 (Topological space).** *A topological space is a set $T$ together with a collection of open subsets $O$ that satisfies the four conditions:*

1. *the empty set $\emptyset$ is in $O$, i.e. $\emptyset \in O$*
2. *$T$ is in $O$, i.e. $T \in O$*
3. *the intersection of a finite number of sets in $O$ is also in $O$, i.e. $\{ S_i \}_{i \leqslant n}$ if $S_i \in O$ then $\bigcap_i S_i \in O$*
4. *the union of an arbitrary number of sets in $O$ is also in $O$, i.e. $\{ S_i \}_{i \in \mathbb{N}}$ if $S_i \in O$ then $\bigcup_i S_i \in O$*

Given a topological space $T$, the Borel $\sigma$-algebra $B$ is defined to be the $\sigma$-algebra generated by the open sets of $T$. The closures of $B$ are numerable union, no more arbitrary union.

For our purposes we are interested in particular topological spaces called metric spaces, where the notion of closeness is substituted by the notion of distance that can say when things are close to each other. From any metric space $(T, d)$ we form a useful $\sigma$-algebra, called the Borel sets. We start with the set of all open balls in $T$, i.e. $B(x, r) = \{x' \in T \mid \mu(x, x') < r\}$ for any $x \in T$ and any $r \in (0, \infty)$. From the open balls, the Borel sets $\mathcal{B}$ are the sets that can be constructed from these open balls by using the $\sigma$-algebra axioms. By using Borel sets the nonmeasurable sets are avoided.

*Example 3.6.* A simple example of $\mathcal{B}$ can be constructed for $\mathbb{R}$. The open balls are just the set of all open intervals, $(x_1, x_2) \subset \mathbb{R}$, for any $x_1, x_2 \in \mathbb{R}$ such that $x_1 < x_2$. ♮

**Definition 3.7 (Borel $\sigma$-algebra).** *Let $(T, d)$ be a metric space. The* Borel $\sigma$-algebra *($\sigma$-field) $B(T)$ is the smallest $\sigma$-algebra in $T$ that contains all open sets of $T$. The elements of $B$ are called the Borel sets of $T$.*

The metric space $(T, d)$ is called separable if it has a countable dense subset, i.e. there are $x_1, x_2, \ldots$ in $T$ such that $\overline{\{x_1, x_2, \ldots\}} = T$. $\overline{A}$ denotes the closure of $A \subset T$.

**Proposition 3.8.** *If $T$ is a separable metric space, then $B(T)$ equals the $\sigma$-algebra, called $A$, generated by the open (or closed) balls of $T$.*

*Proof.* Clearly, $A \subset B$. Let $D$ be a countable dense set in $T$. Let $U \subset X$ be open. For $x \in U$ take $r > 0, r \in Q$ such that $B(x, r) \subset U(B(x, r)$ open or closed ball with center $x$ and radius $r)$ and take $y_x \in D \cap B(x, r/3)$. Then $x \in B(y_x, r/2) \subset B(x, r)$. Set $r_x := r/2$. Then $U = \bigcup \{ B(y_x, r_x) : x \in U \}$, which is a countable union. Therefore $U \in A$. Hence $B \subset A$. □

**Definition 3.9 (Borel probability measures).** *Let $(T, d)$ be a metric space. A finite Borel measure on $T$ is a map $\mu\colon B(T) \to [0, \infty)$ such that*

1. *$\mu(\emptyset) = 0$*
2. *$A_1, A_2, \ldots \in B(T)$ mutually disjoint implie $\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu\left(A_i\right)$*

*$\mu$ is called a* Borel probabiliy measure *if in addition $\mu(T) = 1$.*

The following well known continuity properties will be used many times.

**Proposition 3.10.** *Let $T$ be a metric space and $\mu$ a finite Borel measure on $T$. Let $A_1, A_2, \dots$ be Borel sets.*

1. *If $A_1 \subset A_2 \subset \dots$ and $A = \bigcup_{i=1}^{\infty} A_i$, then $\mu(A) = \lim_{n\to\infty} \mu(A_n)$*
2. *If $A_1 \supset A_2 \supset \dots$ and $A = \bigcap_{i=1}^{\infty} A_i$, then $\mu(A) = \lim_{n\to\infty} \mu(A_n)$*

**Proposition 3.11.** *If $\mu$ is a finite Borel measure on $T$ and $\mathcal{A}$ is a collection of mutually disjoint Borel sets of $T$, then at most countably many elements of $A$ have nonzero $\mu$-measure.*

*Proof.* For $m \geqslant 1$, let $\mathcal{A}_m := \{ A \in \mathcal{A} \mid \mu(A) > 1/m \}$. For any distinct $A_1, \dots, A_k \in \mathcal{A}_m$ we have $\mu(T) \geqslant \mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \mu(A_1) + \dots + \mu(A_k) > k/m$, hence $\mathcal{A}_m$ has at most $m\mu(T)$ elements. Thus $\{ A \in \mathcal{A} \mid \mu(A) > 0 \} = \bigcup_{i=1}^{\infty} \mathcal{A}_m$ is countable. □

## 3.2 Probabilistic systems

In literature concurrency theory performs a study of processes which can be "exactly" exchanged, thus their behaviours are identical. We consider internal actions hidden from the external observers, that may indirectly affect externally visible behaviour. These actions are important when we define behaviour equivalences, in particular two specifications with different internal actions may achieve the same "observable" behaviour and could, thus, be considered equivalent. Since the internal action are formally described by probability distributions, these processes are describe by probabilistic systems. Thus probabilistic models allow to describe essential details which are not captured by the nondeterministic model.

### 3.2.1 General processes

A transition system is an abstract machine that represents either an implementation, i.e., a physical device or software system, or a specification, i.e., a description of the required properties of an implementation. It is one of the first mathematical models for describing the systems and the most common structure used to describe behaviour of processes.

**Definition 3.12.** *A Labelled Transition System is a triple $(W, A, Tr)$ with domain $W$, set of labels $A$, and for each label $a$, a relation $Tr\colon W \times A \to W$ on $W$ called the transition relation.*

The infix notation for relations we write is $s \xrightarrow{a} t$ when $(s, a, t) \in Tr$, in this case we call $t$ a $a$-derivative of s, or sometimes simply a derivative of s. In order to extend Labelled Transition Systems to the probabilistic setting, the main addition that is needed is some mechanism for representing probabilistic choices as well as nondeterministic choices. Thus a probabilistic transition relation $s \xrightarrow{a} \mu$ specifies the probability $\mu$ of moving from one state $s$ to another $t$. When we deal with a general probabilistic system we refer to a Probabilistic Labelled Transition Systems (PLTS), i.e. a transition system with probabilities and labels associated with the transitions.

**Definition 3.13.** *A Probabilistic Labelled Transition Systems (PLTS) is a triple* $(Q, A, Tr)$*, where* $Q$ *is the set of states,* $A$ *is the set of actions,* $Tr\colon Q \times A \rightarrow Disc(Q)$ *is a transition relation. A transition* $(q, a, \mu) \in Tr$ *from a state* $q$ *and action* $a$ *with probability* $\mu$ *is also denoted by* $q \xrightarrow{a} \mu$*.*

We recall that all probabilistic data is internal and no probabilities associated with environment show through. Partial labeled Markov chains are the discrete probabilistic analogs of labeled transition systems, where the final state is governed by a probability distribution and no other indeterminacy.

**Definition 3.14.** *A partial labeled Markov chain (plMc) with a label set* $L$ *is a structure* $(S, \{\, k_l \mid l \in L \,\}, s)$*, where* $S$ *is a countable set of states,* $s$ *is the* $P$ *start state, and* $\forall l \in L . k_l\colon S \times S \rightarrow [0, 1]$ *is a transition function such that* $\forall s \in S. \sum_t k_l(s, t) \leqslant 1$*.*

We could have alternatively presented a plMc as a structure $(S, \{\, k_l \mid l \in L \,\}, \mu)$, where $\mu$ is an initial distribution on $S$. Given a plMc with initial distribution $P$, one can construct an equivalent plMc with initial state $Q$ as follows. $S_Q = S_P \cup \{u\}$ where $u$ is a new state not in $S_P$. $u$ will be the start state of Q. $k_l^Q(s, t) = k_l^P(s, t)$ if $s, t \in S_P$; $k_l^Q(s, u) = 0$, and $k_l^Q(u, t) = \sum k_l^P(s, t)\mu^P(s)$. We will freely move between the notions of initial state and initial distribution. For example, when a transition P on label $l$ occurs in a plMc $P$, there is a new initial distribution given by $\mu'(t) = \sum k_l(s, t) \times \mu(s)$.

When the system interacts with the environment and in addition to the probabilistic moves, non-deterministic choices are possible. Such choices are captured by Markov Decision Processes (MDP), which extend Markov chains with non-determinism.

**Definition 3.15 (MDP).** *A Markov Decision Process (MDP) is* $(S, A, P, R(\cdot, \cdot), \gamma)$ *such that*

- $S$ *is a finite set of* states
- $A$ *is a finite set of* actions
- $P_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ *is the probability that the action* $a$ *in states at time* $t$ *will lead to state* $s'$ *at time* $t + 1$
- $R_a(s, s')$ *is the immediate reward received after transitioning from state* $s$ *to state* $s'$*, due to action* $a$
- $\gamma \in [0, 1]$ *is the discount factor, which represents the difference in importance between future rewards and present rewards.*

A MDP is a submodel of a game structure, where a game is any situation with the following three aspects. 1) There is a set of participants, whom we call the players. 2) Each player has a set of options for how to behave; we will refer to these as the players possible strategies. 3) For each choice of strategies, each player receives a payoff that can depend on the strategies selected by everyone. The payoffs will generally be numbers, with each player preferring larger payoffs to smaller payoffs. A game structure $G$ is a MDP if only one of the two players has a choice of moves. For $i \in \{1, 2\}$, we say that a structure is an $i$-MDP if for all $s \in S$ the move assignments of not such that $i$ $|\mathit{\Gamma}_{j \neq i}(s)| = 1$. For MDPs, we omit the (single) move of the player without a choice of moves, and write $\delta(s, a)$ for the transition function. Game structure generalises many common structures in computer science, from transition systems, to Markov chains and Markov decision processes. We consider games with simultaneous moves, where the players randomise their moves at each round. Intuitively, the adversary cannot play the exact winning move in response to each the individual move played. The players play not a single move called *pure move*, rather a probability distribution over the available moves at a state called *mixed move*. Before to give the definition of game structure, we introduce some notation related to basic probabilistic concept. For a finite set $A$, let $\mathrm{Dist}(A) = \{ \mu \colon A \to [0, 1] \mid \sum_{a \in A} \mu(a) = 1 \}$ denotes the set of probability distributions over $A$. We denote by $D_i(s) = Dist(\mathit{\Gamma}_i(s))$ the set of mixed moves available to player $i \in \{1, 2\}$ in the state $s$. We extend the transition function to mixed moves. Given a state $s$ and $x_1 \in D_1(s), x_2 \in D_2(s)$, we write $\delta(s, x_1, x_2)$ for the next-state probability distribution induced by the mixed moves $x_1$ and $x_2$, defined for all $t \in S$ by $\delta(s, x_1, x_2)(t) = \sum_{a_1 \in \mathit{\Gamma}_1(s)} \sum_{a_2 \in \mathit{\Gamma}_2(s)} \delta(s, a_1, a_2)(t) x_1(a_1) x_2(a_2)$.

A *valuation* over $A$ is a function $f \colon A \to [0, 1]$ that associates to every element $s \in A$ a value $0 \leqslant f(s) \leqslant 1$. The set of all valuations is $\mathcal{F}$, where for $f, g \in \mathcal{F}$ we write $f \leqslant g$ if and only if $f(s) \leqslant g(s)$ at all $s \in A$. We recall that $\mathcal{F}$ under $\leqslant$ forms a complete lattice.

**Definition 3.16 (Game structure [12]).** *Given a fixed finite set $\mathit{\Upsilon}$ of observation variables. A (two-player, concurrent)* game structure $G = \langle S, [\cdot], Moves, \mathit{\Gamma}_1, \mathit{\Gamma}_2, \delta \rangle$ *consists of the following components:*

- *a finite set $S$ of* states
- *a variable interpretation $[\cdot] \colon \mathit{\Upsilon} \times S \to [0, 1]$, which asociates with each variable $v \in \mathit{\Upsilon}$ a valuation $[v]$*
- *a finite set $Moves$ of moves*
- *two move assignments $\mathit{\Gamma}_1, \mathit{\Gamma}_2 \colon S \to \{2^{Moves} \setminus \emptyset\}$. For $i \in \{1.2\}$, the assignment $\mathit{\Gamma}_i$ associates with each state $s \in S$ the nonempty set $\mathit{\Gamma}_i(s) \subseteq Moves$ of moves available to player $i$ at state $s$*
- *a probabilistic transition function $\delta \colon S \times Moves \times Moves \to Dist(S)$, that gives the probability $\delta(s, a_1, a_2)(t)$ of a transition from $s$ to $t$ when player $1$ plays move $a_1$ and player $2$ plays move $a_2$.*

Given a valuation $f \in F$, a state $s \in S$, and two mixed moves $x_1 \in D_1(s), x_2 \in D_2(s)$, we define the *expectation of* $f$ from $s$ under $x_1, x_2$ as $\mathbb{E}_s^{x_1, x_2}(f) = \sum_{t \in S} \delta(s, x_1, x_2)(t) f(t)$. For a game structure G, for $i \in \{1, 2\}$ we define the *valuation transformer* $\mathrm{Pre}_i \colon F \to F$ by, for all $f \in F$ and $s \in S$,

$\mathrm{Pre}_i(f)(s) = \sum_{x_i \in D_i(s)} \inf_{x_{\neg i} \in D_{\neg i}(s)} \mathbb{E}_s^{x_1,x_2}(f)$. Intuitively, $\mathrm{Pre}_i(f)(s)$ is the maximal expectation player i can achieve of f after one step from s: this is the classical one-day or next-stage operator of the theory of repeated games.

The use of mixed moves allows players to win with probability 1 games that they would lose, i.e. win with probability 0, if restricted to playing moves without simultaneity. Inserting the probabilities the question of winning a game is thus a probabilistic one: what is the maximal probability whereby a player can be guaranteed of winning, regardless of how the other player plays? This probability is known, in brief, as the winning probability.

### 3.2.2 Probabilistic automata

The analysis of distributed and concurrent systems developed the mathematical model of Transition System (TS), that describes the effect of operations (called transitions) on the systems state. An extension of TS explicit labels the transitions of the system by actions, such that the execution of an action is the result of a change of a state. The extended TS is called a Labelled Transition System (LTS). A TS is used to describe the potential behaviours of discrete systems.

To model and study randomized distributed algorithms into the concurrency theory is used the Probabilistic Automata model. Finite state automata differ from TS by 1) the sets of states and the set of transition are necessary finite, or even countable; 2) start state or finite state are given. An automata is a state machine with labelled steps, call also transitions. Its action describes he interface with the external environment by specifying which actions model events that are visible from the external environment and which ones model internal events. A probabilistic automata differ from an automaton in that the action and the next stage of a given transition are chosen probabilistically.

**Definition 3.17 (Probabilistic Automaton).** *A Probabilistic Automaton (PA) is a tuple $(S, \bar{s}, A, D)$ where $S$ is a set of states, $\bar{s} \in S$ is the start state, $A$ is a set of actions, and $D \subseteq S \times A \times Disc(S)$ is a transition relation. The set of actions $A$ is further partitioned into three sets I, O, H of input, output and internal (hidden) actions, respectively.*

PA model provides the tools to relate executions of different systems. Simulations and bisimulations allow us to compare the computations of two systems and to say if they behave in the same way or if their behaviours are not similar.

## 3.3 Bisimulation

A classical method to simplify the study of a Markov chain is to a simpler Markov chain consists of founding an equivalence relation $\equiv$ on states of the chain such that the simplified transition function $\mu\colon X \to P(X)$, defined by $\equiv$, does not change the probabilities of the transitions of the original chain. If we consider a Markov labelled system, two states are equivalent $s_1 \equiv s_2$ if we cannot distinguish $s_1$ from $s_2$, i.e. there is no difference between the probability of pass through $s_1$ toward $C$ or the probability of pass through $s_2$ toward $C$. This notion used by [37], it is called *bisimulation* when $X$ is a finite space.

The most studied form of behavioural equality for processes in concurrency is the bisimulation equality, called also bisimilarity. Some of reasons for which it is widely used are the followings.

(a) Bisimilarity is accepted as the finest behavioural equivalence one would like to impose on processes.
(b) The bisimulation proof method is exploited to prove equalities among processes. This occurs even when bisimilarity is not the behavioural equivalence chosen for the processes. For instance, one may be interested in trace equivalence and yet use the bisimulation proof method since bisimilarity implies trace equivalence.
(c) The efficiency of the algorithms for bisimilarity checking and the compositionality properties of bisimilarity are exploited to minimise the state-space of processes.
(d) Bisimilarity, and variants of it such as similarity, are used to abstract from certain details of the systems of interest. For instance, we may want to prove behavioural properties of a server that do not depend on the data that the server manipulates. Further, abstracting from the data may turn an infinite-state server into a finite one.

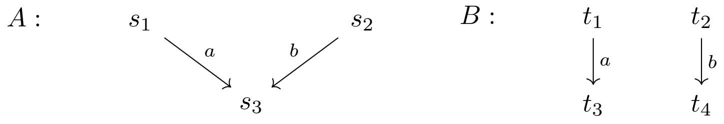
Bisimulation has been derived through refinements of notions of morphism between algebraic structures. We consider automata where the basic description of a behaviour is a single transition denotes the execution of an action. Here a process is modelled as a sequence of transitions from (a set of) an initial state $s_0$ and a set of final state $O$, these states belong to a finite set $Q$ and the finite set of transitions between states is $Tr$. A process can also be described as an element of an algebra of regular expressions [38], by using the axioms and equational reasoning we can perform calculations with processes. The simplest model of a process transition is an input/output function, where a values given in input is processes to obtain the value of the output. This transition function is a partial one and it is denoted by $Tr_a\colon Q \to Q$; similarly, we denote as $O_a\colon Q \to O$ the output function. The modern parallel and distributed systems are based on interactions between systems during the execution, these interactions are at the base of concurrency theory. We consider two automata $A, B$ as in [3], a homomorphism from the automaton A to the automaton B is a surjective function $F \colon Q^A \to Q^B$ such that for all input $a$:

- $F \circ Tr_a^A \subseteq Tr_a^B \circ F$, that is the condition on the states
- $O_a^A \subseteq O_a^B \circ F$, that is the condition on the outputs.

We rewrite these constraints as operational constraints on automata, where $s \xrightarrow[b]{a} t$ denotes the transition from state $s$ and input $a$ produces the output $b$ and evolves into the state $t$. Assuming for simplicity that $O_a^A$ and $Tr_a^A$ are undefined exactly on the same points, the two conditions above become:
for all $s, s' \in Q^A$, if $s \xrightarrow[b]{a} s'$ then also $F(s) \xrightarrow[b]{a} F(s')$.
If there is a homomorphism from $A$ to $B$ then automaton $B$ covers automaton $A$, i.e., $B$ can do, state-wise, at least all the transitions that $A$ does. That is, there is a total function $\varphi$ from the states of $A$ to the states of $B$ such that, for all states $s$ of $A$, all translations performed by $A$ when started in $s$ can also be performed by $B$ when started in $\varphi(s)$. The implication becomes stronger if one uses weak homomorphisms, which are obtained by relaxing the functional requirement of homomorphism into a relational one. Thus a weak homomorphism is a total relation $R \colon Q^A \times Q^B$ such that for all input $a$:

1. $Tr_a^A \circ R^{-1} \subseteq R^{-1} \circ Tr_a^B$, that is the condition on the states
2. $O_a^A \circ R^{-1} \subseteq O_a^B$, that is the condition on the outputs.

We give the operational interpretation: whenever $s\,R\,t$ and $s \xrightarrow[b]{a} s'$ hold in A, then there is $t'$ such that $t \xrightarrow[b]{a} t'$ holds in B and $s'\,R\,t'$.
As homomorphisms, so weak homomorphisms imply that automaton $B$ covers automaton $A$. The result for weak homomorphism is stronger as the homomorphisms are strictly included in the weak homomorphisms. We consider two automata $A, B$, some states $\{s_i\} \in A$ and $\{t_i\} \in B$ and the transitions $s_1 \xrightarrow{a} s_3, s_2 \xrightarrow{b} s_3, t_1 \xrightarrow{a} t_3, t_2 \xrightarrow{b} t_4$ showed in the following diagram.



We cannot establish a homomorphism $F$ from $A$ to $B$, since a homomorphism must be surjective and the functional requirement prevents us from relating $s_3$ with both $t_3$ and $t_4$. $F$ is constructed with the following relations

$$s_1 \xrightarrow{a} s_3 \implies F(s_1) = t_1 \xrightarrow{a} F(s_3) = t_3$$
$$s_2 \xrightarrow{b} s_3 \implies F(s_2) = t_1 \xrightarrow{b} t_4 \neq F(s_3) = t_3$$

By contrast, a weak homomorphism $R$ exists and relates $s_1$ with $t_1$, $s_2$ with $t_2$, and $s_3$ with both $t_3$ and $t_4$, as shown in the following.

$$s_1\, R\, t_1 \text{ and } s_1 \xrightarrow{a} s_3 \implies t_1 \xrightarrow{a} t_3 \text{ and } s_3\, R\, t_3$$

$$s_2\, R\, t_4 \text{ and } s_2 \xrightarrow{b} s_3 \implies t_2 \xrightarrow{a} t_4 \text{ and } s_3\, R\, t_4.$$

### 3.3.1 Classical bisimulation

Intuitively, two processes should be equivalent if they cannot be distinguished by interacting with them. The definition of bisimulation relations establish step-by-step two-directional correspondences of behaviours between two states or systems. In the following we give the formal definition starting with the definition of simulation, which are half bisimulations.

We let R range over relations on sets, for example if $\mathcal{P}$ denotes the powerset construct, then a relation R on a set W is an element of $\mathcal{P}(W \times W)$. We use the infix notation for relations $(s,t) \in R$ by means $s\, R\, t$.

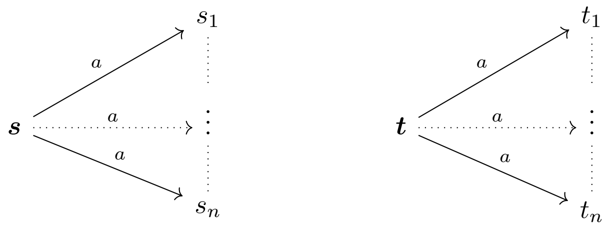
**Definition 3.18 (Strong simulation).** *A binary relation $R$ on the states of an LTS is a* strong simulation *if $s\, R\, t$ implies that for all $s_1$ with $s \xrightarrow{a} s_1$ there is $t_1$ such that $t \xrightarrow{a} t_1$ and $s_1 R t_1$. Similarity is the union of all simulations.*

In the diagrams of the thesis we represent the $R$-relation between states $s$ and $t$ such that $s\, R\, t$ with an sawtoothed right arrow $\rightsquigarrow$ from $s$ to $t$, with the meaning of "$s$ can be substituted with $t$". In Diagram 3.1 the transition applied to $s$ with the action $a$ is deterministic, i.e. there is at most one state $s_1$ such that $s \xrightarrow{a} s_1$. $t$ simulates $s$, thus the action $a$ on $t$ induces the transition $t \xrightarrow{a} t_1$ such that $t_1$ simulates $s_1$.

$$\begin{tikzcd}
s \arrow[r, rightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & t \arrow[d, "a"] \\
s_1 \arrow[r, rightsquigarrow, "\boldsymbol{R}"] & t_1
\end{tikzcd}$$

(D.3.1)

In the left hand side of Diagram 3.2 we represent a nondeterministic transition with a set of arrows from state $s$ and action $a$, which leads to a set of states $\{s_i\}_{1\leqslant i\leqslant n}$. Each arrow represents a possible transition which leads to a single state $s_i$. In the right hand side of Diagram 3.2 we represent the nondeterministic transition with action $a$ from the state $t$.



(D.3.2)

$t$ simulates $s$, thus each $s_i$ of a possible transition has to be simulated by an outcome $t_i$ of a transition $t \xrightarrow{a} t_i$, as shown in Diagram 3.3. We highlight the strict constraint of the definition of simulation: the number of possible outcomes $\{s_i\}_{1\leqslant i\leqslant n}$ is equal to the number of possible outcomes $\{t_i\}_{1\leqslant i\leqslant n}$.

$$\begin{tikzcd}
s \arrow[r, rightsquigarrow, "\boldsymbol{R}"] \arrow[d, bend left, "a"] \arrow[dd, dotted, bend right, "a"'] \arrow[ddd, bend right=60, "a"'] & t \arrow[d, bend right, "a"'] \arrow[dd, dotted, bend left, "a"] \arrow[ddd, bend left=60, "a"] \\
s_1 \arrow[r, rightsquigarrow, "\boldsymbol{R}"] & t_1 \\
\cdots \arrow[r, rightsquigarrow, "\boldsymbol{R}"] & \cdots \\
s_n \arrow[r, rightsquigarrow, "\boldsymbol{R}"] & t_n
\end{tikzcd} \tag{D.3.3}$$

When the relation $R$ is an equivalence, $s$ simulates $t$ and the converse too.

**Definition 3.19 (Strong bisimulation).** *An equivalence binary relation $R$ on the states of an LTS is a* strong bisimulation *if whenever $s\,R\,t$*

- *for all $s_1$ with $s \xrightarrow{a} s_1$, there is $t_1$ such that $t \xrightarrow{a} t_1$ and $s_1 R\, t_1$*
- *the converse, i.e. for all $t_1$ there exists $s_1$ such that $t_1 R\, s_1$.*

The notation of strong bisimulation is represented by a double sawtoothed arrow as shown in Diagram 3.4.

$$\begin{tikzcd}
s \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & t \arrow[d, "a"] \\
s_1 \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] & t_1
\end{tikzcd} \tag{D.3.4}$$

Bisimilarity is the union of all bisimulations, i.e. the maximum probabilistic bisimulation.

**Definition 3.20 (Bisimilarity).** *Given a probabilistic system $(Q, A, Tr)$ two state $s, t$ are bisimilar and denoted by $s \sim t$ if and only if there exists a strong bisimulation $R$ such that $s\,R\,t$.*

We represent bisimilarity in Diagram 3.5.With respect to Diag. 3.4 we have add the double arrow labelled $\sim$, which is nearly a circle since it is the last step of the definition.

$$\begin{tikzcd}
s \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"] \arrow[r, leftrightarrow, dash dot, bend right=90, looseness=3, "\sim"'] & t \arrow[d, "a"'] \\
s_1 \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] & t_1
\end{tikzcd} \tag{D.3.5}$$

The (bi)simulation can be relaxed to be a weak (bi)simulation.

**Definition 3.21 (Weak simulation).** *A binary relation $R$ on the states of an LTS is a* weak simulation *if $s\,R\,t$ implies that for all $s_1$ with $s \xrightarrow{a} s_1$ there is $t_1$ such that $t \overset{a}{\twoheadrightarrow} t_1$ and $s_1 R\, t_1$.*

If two states $s$ and $t$ are bisimilar and there is an action leading from $s$ to the state $s'$, then there must exist a state $t'$ such that there is some number, possibly zero, of actions leading from $t$ to $t'$, that we denote by $\overset{a}{\twoheadrightarrow}$. In Diagram 3.6 we represent this new notion.

$$\begin{tikzcd}
s \arrow[r, squiggly, "\boldsymbol{R}"] \arrow[dd, "a"'] & t \arrow[d, two heads, "a"] & \\
& \vdots \arrow[d, two heads, "a"] & \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} 0 \text{ or } n \\
s' \arrow[r, squiggly, "\boldsymbol{R}"] & t' &
\end{tikzcd} \tag{D.3.6}$$

We define the weak bisimulation as a bidirectional relation.

**Definition 3.22 (Weak bisimulation).** *A* weak bisimulation *is a binary relation $R$ on the set of processes such that for all $s, t$, if $s\,R\,t$ then*

- *for all action $a$ and state $s'$ if $s \xrightarrow{a} t'$, then $t \overset{a}{\twoheadrightarrow} t'$ and $s'\,R\,t'$*
- *for all action $a$ and state $t'$ if $t \xrightarrow{a} t'$, then $s \overset{a}{\twoheadrightarrow} s'$ and $t'\,R\,s'$.*

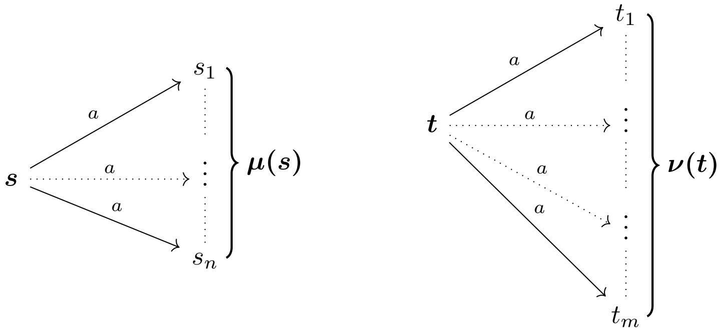
Since in a weak simulation a single transition of a state is simulated by more than one transition of the other state, then we cannot generate a diagram with bidirectional arrows as in Diag. 3.4 and thus we generate two diagrams in Diagram 3.7.

$$\begin{tikzcd}
s \arrow[r, squiggly, "\boldsymbol{R}"] \arrow[dd, "a"'] & t \arrow[d, two heads, "a"] & & & & s \arrow[d, two heads, "a"] & t \arrow[l, squiggly, "\boldsymbol{R}"'] \arrow[dd, "a"'] \\
& \vdots \arrow[d, two heads, "a"] & \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} 0 \text{ or } n & \text{and} & 0 \text{ or } n \left\{\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right. & \vdots \arrow[d, two heads, "a"] & \\
s' \arrow[r, squiggly, "\boldsymbol{R}"] & t' & & & & s' & t' \arrow[l, squiggly, "\boldsymbol{R}"']
\end{tikzcd} \tag{D.3.7}$$

**Definition 3.23 (Weak bisimularity).** *Two states $s$ and $t$ of a probabilistic system are* weakly bisimilar*, denoted $s \simeq t$, if there exists a bisimulation $R$ such that $s\,R\,t$.*

### 3.3.2 Probabilistic bisimulation

The extension of strong bisimulation and strong simulation to the probabilistic framework presents a problem, which is due to the fact that a probabilistic transition leads to a probability distribution over states rather than to a single state. This problem is represented in Diagram 3.8, in the left hand side the probabilistic transition from the state $s$ with action $a$ leads to a set of states described by the probabilistic distributions $\mu(s)$. In the right hand side $\nu(t)$ is the probabilistic distributions of the states $\{t_j\}_{1 \leqslant j \leqslant m}$, which are the results of the transition $t \xrightarrow{a} \nu(t)$. We note that the number of the resulting states of two different transitions can be different, as in the diagram where $n > m$.



(D.3.8)

Thus, to extend the notion of (bi)simulation to the probabilistic framework a relation over states needs to be lifted to distributions over states.

**Definition 3.24 (Lifting).** *Given a relation $R \subseteq X \times Y$, the* lifting *of $R$ is the relation $L(R)\colon Disc(X) \to Disc(Y)$ such that there exists a weighting function $w : X \times Y \to [0,1]$ satisfying the following properties, for $\mu \in Disc(X)$ and $\nu \in Disc(Y)$*

1. $w(s,t) > 0$ *implies* $s\,R\,t$
2. $\sum_{s \in X} w(s,t) = \mu(t)$
3. $\sum_{t \in Y} w(s,t) = \nu(s)$

*We denote the discrete probability measures in lifting relation by $\mu\,L(R)\,\nu$.*

An alternative definition of lifting given in a more probabilistic style is the following. $\mu\,L(R)\,\nu$ if and only if there exists a joint measure $w$ with marginal measures $\mu$ and $\nu$ such that the support of $w$ is included in $R$. If $R$ is an equivalence relation, then $\mu\,L(R)\nu$ if and only if, for each equivalence class C of $R$, $\mu(C) = \nu(C)$. The notation of the lifting relation in the original version is $\sqsubseteq_R$ ( [57,58]).

In the following we show some properties that we will use in the rest of the thesis.

**Proposition 3.25.** *Let $R$ is a relation on $Q$ and $L(R)$ is the lifting relation between measures induced by $R$, $R = \emptyset$ if and only if $L(R) = \emptyset$.*

*Proof.*

($\Rightarrow$) For hypothesis we assume $R = \emptyset$ and $L(R) \neq \emptyset$. For definition of $L(R)$ there exist two measures $\mu, \nu \in \mathrm{Disc}(Q)$ such that $\mu\, L(R)\, \nu$, therefore there exists a weighting function $w\colon Q \times Q \to \mathbb{R}$ with marginal distributions $\mu$ and $\nu$. Since $\mu$ is a probabilistic measure, then $\sum_{s\in Q} \mu(s) = 1$ and in particular there exists a state $s \in Q$ such that $\mu(s) > 0$. We calculate the value of the latter using a properties of $w$, i.e. $\mu(s) = \sum_{t\in Q} w(s,t)$. Since $w(s,t) > 0$, this implies that there exists a $t \in Q$ which is, for a second properties of $w$, in relation $s\,R\,t$. This is an absurd since $R$ is empty, so we have shown that $R = \emptyset \implies L(R) = \emptyset$.

($\Leftarrow$) We recall the definition of a Dirac measure, that is a measure $\delta_x$ on a set $X$ defined for a given $x \in X$ and any measurable set $A \subseteq X$ by $\delta_x(A) = 1$ if $x \in A$ and $\delta_x(A) = 0$ if $x \notin A$. If $R \neq 0$, then there exist $s, t \in Q$ such that $s\,R\,t$. We define two Dirac measures $\delta_s, \delta_t$ and we want to obtain $\delta_s\, L(R)\, \delta_t$. We define a weighting function $w$ as, for every $x, y \in Q$

$$w(x,y) = \begin{cases} 1 & \text{if } x = s \text{ and } y = t \\ 0 & \text{otherwise} \end{cases}$$

We have $w(s,t) = 1$ and for the other cases we have 0, i.e. $\sum_{x,y\in Q. x\neq s \wedge y \neq t} w(x,y) = 0$, except for $\sum_{y\in Q} w(s,y) = \delta_t$ and $\sum_{y\in Q} w(x,t) = \delta_s$. Thus the axioms 3. and 4. are verified above, the function is non-negative and thus also the axiom 1. is verified. To verify the axiom 2. we check when $w(x,y) > 0$, the only case is $w(s,t) = 1$ for $x = s, y = t$ and thus $s\,R\,t$.

□

**Proposition 3.26 (Reflexive lifting).** *If $R$ is reflexive, then $L(R)$ is reflexive.*

*Proof.* $R$ is reflexive and for each $x \in Q$ we have $x\,R\,x$. Let $\mu \in \mathrm{Disc}(Q)$ is a probabilistic measure, on which we define the following function $w\colon Q\times Q \to \mathbb{R}$ for each $x, y \in Q$

$$w(x,y) = \begin{cases} \mu(x) & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

It verifies the weighting function axioms:

1. $\mu$ is a measure, i.e. with outcomes in the interval $[0,1]$, and thus $w$ is non-negative
2. for each $x, y \in Q$ if $w(x,y) > 0$, then $x = y$. Since $R$ is reflexive, we have $x\,R\,y$
3. for each $x \in Q$ we have $\sum_{y\in Q} w(x,y) = w(x,x) = \mu(x)$; in the other cases $y \neq x$ implies $w(x,x) = 0$
4. As above, for each $y \in Q$ we have $\sum_{x\in Q} w(x,y) = w(y,y) = \mu(y)$

□

**Proposition 3.27 (Symmetric lifting).** *If $R$ is symmetric, then $L(R)$ is symmetric.*

*Proof.* Let $R$ be a symmetric relation on $Q$ and $\mu_1 \in \mathrm{Disc}(Q_1), \mu_2 \in \mathrm{Disc}(Q_2)$ be two measures such that $\mu_1 L(R) \mu_2$. This means that there exists a $w\colon Q_1 \times Q_2 \to [0,1]$ which is a weight function. $w$ verifies the lifting properties:

- for each $u \in Q_1$ and $v \in Q_2$ we have $w(u,v) > 0$ that implies $u\,R\,v$
- for each $u \in Q_1$ and $\sum_{v \in Q_2} w(u,v) = \mu_1(u)$
- for each $v \in Q_2$ and $\sum_{u \in Q_1} w(u,v) = \mu_2(v)$

We define $w'\colon Q_1 \times Q_2 \to [0,1]$ as $w'(u,v) = w(v,u)$. It is easy to show that $w'$ verifies the lifting properties:

- for each $u \in Q_1, v \in Q_2$ we have $w'(u,v) > 0$ that implies $w(v,u) > 0$, thus $v\,R\,u$ and since R is symmetric, we have that $u\,R\,v$
- for each $u \in Q_1$ and $\sum_{v \in Q_2} w'(u,v) = \sum_{v \in Q_2} w(v,u) = \mu_2(u)$
- for each $v \in Q_2$ and $\sum_{u \in Q_1} w'(u,v) = \sum_{u \in Q_1} w(v,u) = \mu_1(v)$

This implies that $\mu_2 L(R) \mu_1$. ☐

**Proposition 3.28 (Transitive lifting).** *If $R$ is transitive, then $L(R)$ is transitive.*

To show this proposition we have to show the following property.

**Proposition 3.29.** *Let $R$ and $S$ be two relations and $\mu_1, \mu_2, \mu_3$ be three probability measures. If $\mu_1 L(R) \mu_2$ and $\mu_2 L(R) \mu_3$, then $\mu_1 L(R \circ S) \mu_3$.*

*Proof.* Let $R, S$ be two relations from $Q_1$ to $Q_2$ and from $Q_2$ to $Q_3$, respectively. Let $\mu_1 \in \mathrm{Disc}(Q_1)$, $\mu_2 \in \mathrm{Disc}(Q_2)$, $\mu_3 \in \mathrm{Disc}(Q_3)$ be three probability measures such that $\mu_1 L(R) \mu_2$ and $\mu_2 L(S) \mu_3$. This implies that there exist $w_r\colon Q_1 \times Q_2 \to [0,1]$ and $w_s\colon Q_2 \times Q_3 \to [0,1]$ such that

- for each $u \in Q_1, v \in Q_2$ we have $w_r(u,v) > 0$ that implies $(u,v) \in R$
- for each $u \in Q_1$ we have $\sum_{v \in Q_2} w_r(u,v) = \mu_1(u)$
- for each $v \in Q_2$ we have $\sum_{u \in Q_1} w_r(u,v) = \mu_2(v)$
- for each $u \in Q_2, v \in Q_3$ we have $w_s(u,v) > 0$ that implies $(u,v) \in S$
- for each $u \in Q_2$ we have $\sum_{v \in Q_3} w_s(u,v) = \mu_2(u)$
- for each $v \in Q_3$ we have $\sum_{u \in Q_2} w_s(u,v) = \mu_3(v)$

We define $w_{rs}\colon Q_1 \times Q_3 \to [0,1]$ as $w_{rs} = \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_r(u,t) w_s(t,v)}{\mu_2(t)}$. $w_{rs}$ is a weight function:

- for each $u \in Q_1$ and $v \in Q_3$, since $w_{rs}(u,v) > 0$, we have that the value of the sum defining $w_{rs}(u,v)$ via the state $t$ is positive and thus there exists $q \in Q_2$ such that $\mu_2(q) \neq 0$ and $\frac{w_r(u,q) w_s(q,v)}{\mu_2(q)} > 0$. Since $\mu_2(q)$ is a probability measure, it follows that $\mu_2(q) > 0$ and thus $w_r(u,q) w_s(q,v) > 0$ that implies $w_r(u,q) > 0$ and $w_s(q,v) > 0$. Hence we have that $u\,R\,q$ and $q\,R\,v$ and thus $u\,R \circ S\,v$

- for each $u \in Q_1$ we have

$$\begin{aligned}
\sum_{v \in Q_3} w_{rs}(u,v) &= \sum_{v \in Q_3} \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_r(u,t) w_s(t,v)}{\mu_2(t)} \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \sum_{v \in Q_3} \frac{w_r(u,t) w_s(t,v)}{\mu_2(t)} \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_r(u,t)}{\mu_2(t)} \sum_{v \in Q_3} w_s(t,v) \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_r(u,t)}{\mu_2(t)} \mu_2(t) \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} w_r(u,t) \\
&= \sum_{t \in Q_2} w_r(u,t) \\
&= \mu_1(u)
\end{aligned}$$

  We can remove the condition on $\mu_2(t) \neq 0$ from the summation since by definition of weighting function, if $\mu_2(t) = 0$, the $\sum_{u \in Q_1} w_r(u,t) = 0$ and thus for each $u \in Q_1$ we obtain $w_r(u,t) = 0$
- for each $v \in Q_3$ we have

$$\begin{aligned}
\sum_{u \in Q_1} w_{rs}(u,v) &= \sum_{u \in Q_1} \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_r(u,t) w_s(t,v)}{\mu_2(t)} \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \sum_{u \in Q_1} \frac{w_r(u,t) w_s(t,v)}{\mu_2(t)} \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_s(t,v)}{\mu_2(t)} \sum_{u \in Q_1} w_r(u,t) \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} \frac{w_s(t,v)}{\mu_2(t)} \mu_2(t) \\
&= \sum_{t \in Q_2, \mu_2(t) \neq 0} w_s(t,v) \\
&= \sum_{t \in Q_2} w_s(t,v) \\
&= \mu_3(v)
\end{aligned}$$

  We can remove the condition on $\mu_2(t) \neq 0$ from the summation since by definition of weighting function, if $\mu_2(t) = 0$, the $\sum_{v \in Q_3} w_r(t,v) = 0$ and thus for each $v \in Q_3$ we obtain $w_s(t,v) = 0$

This implies that $w_{rs}$ is a weighting function from $\mu_1$ to $\mu_3$ and thus $\mu_x L(R \circ S)\mu_3$. ☐

*Proof. Transitivity of $L(R)$.* Let $R$ be a transitive relation on $Q$ and let $\mu_1, \mu_2, \mu_3 \in \mathrm{Disc}(Q)$ be three probability measures such that $\mu_1 L(R)\mu_2$ and $\mu_2 L(R)\mu_3$. By the Property 6, we have that $\mu_1 L(R \circ R)\mu_1$. If $R \circ R \subseteq R$, then by Property 3 we have that $\mu_1 L(R)\mu_3$, as required. So, let $x, z \in Q$ be two states such that $(x, z) \in R \circ R$. By definition of composition, it follows that there exists $y \in Q$ such that $(x, y) \in R$ and $(y, z) \in R$. Since $R$ is transitive, we have that $(x, z) \in R$ and thus $R \circ R \subseteq R$. ☐

We give the definition of probabilistic simulation.

**Definition 3.30 (Strong probabilistic simulation).** *A* strong probabilistic simulation *between two probabilistic automata $A_1$ and $A_2$, without common states, is a relation $R \subseteq Q(A_1) \times Q(A_2)$ such that*

- *each start state $\bar{s}$ of $A_1$ is in relation with at least one start state $\bar{t_i}$ of $A_2$, i.e., $\bar{s}\, R\, \{\bar{t_i}\}_i$*
- *for each pair of states $s\, R\, t$ and each transition $s \xrightarrow{a} \mu$ of $A_1$, there exists a transition $t \xrightarrow{a} \nu$ of $A_2$ such that $\mu\, L(R)\, \nu$.*

The main difference with classical simulation in Def. 3.18 is the use of probabilistic distributions for describing the resulting states, as shown in Diag. 3.8. We update Diag. 3.3 inserting the distributions in Diagram 3.9. The latter new diagram covers also the update of Diag. 3.1, since $\mu(s)$ includes all the possible resulting states. Now the relation between the two set of reached states, i.e., $\{s_i\}_i$ and $\{t_i\}_i$, is described by the lifting relation (Def. 3.24) instead of the simple given relation $R$. In Diagram 3.9 this is represented by an arrow with label $L(R)$ instead of $R$.

$$\begin{tikzcd}
s \arrow[r, rightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & t \arrow[d, "a"] \\
\mu \arrow[r, rightsquigarrow, "\boldsymbol{L(R)}"] & \nu
\end{tikzcd} \tag{D.3.9}$$

We consider $R$ to be an equivalence relation between states, thus $s\, R\, t$ and also $t\, R\, s$.

**Definition 3.31 (Strong probabilistic bisimulation).** *A* strong probabilistic bisimulation *between two probabilistic automata $A_1$ and $A_2$, without common states, is an equivalence relation $R$ over states $Q(A_1) \cup Q(A_2)$ such that*

- *each start state $\bar{s}$ of $A_1$ is in relation with at least one start state $\bar{t_i}$ of $A_2$ and vice versa, i.e., $\bar{s}\, R\, \{\bar{t_i}\}_i$ and $\bar{t}\, R\, \{\bar{s_i}\}_i$*
- *for each pair of states $s\, R\, t$ and each transition $s \xrightarrow{a} \mu$ of $A_1$, there exists a transition $t \xrightarrow{a} \nu$ of $A_2$ such that $\mu\, L(R)\, \nu$ and vice versa.*

We represent the probabilistic bisimulation relation on $R$ in Diagram 3.10. Since lifting relation preserves the equivalence property, then we draw the more precise diagram Diag. 3.11 where both $R$ and $L(R)$ are represented by a double arrow.

$$\begin{tikzcd} s \arrow[r, rightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & t \arrow[d, "a"] \\ \mu \arrow[r, rightsquigarrow, "\boldsymbol{L(R)}"] & \nu \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} t \arrow[r, rightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & s \arrow[d, "a"] \\ \nu \arrow[r, rightsquigarrow, "\boldsymbol{L(R)}"] & \mu \end{tikzcd} \tag{D.3.10}$$

$$\begin{tikzcd} s \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"'] & t \arrow[d, "a"] \\ \mu \arrow[r, leftrightsquigarrow, "\boldsymbol{L(R)}"] & \nu \end{tikzcd} \tag{D.3.11}$$

**Definition 3.32 (Strong probabilistic bisimilarity).** *Two states $s, t$, that belong to probabilistic automata $A_1$ and $A_2$, are* bisimilar $s \overset{L}{\sim} t$, *iff there exists a strong bisimulation $R \subseteq Q \times Q$ such that $s\,R\,t$ and $\mu\,L(R)\,\nu$.*

We represent bisimilarity in Diagram 3.12.With respect to Diag. 3.10 we have add the double arrow labelled $\overset{L}{\sim}$, which is nearly a circle since it is the last step of the definition.

$$\begin{tikzcd} s \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"'] \arrow[r, leftrightarrow, dash dot, bend left=40, "\overset{\boldsymbol{L}}{\sim}"] \arrow[d, "a"] & t \arrow[d, "a"'] \\ \mu \arrow[r, leftrightsquigarrow, "\boldsymbol{L(R)}"] & \nu \end{tikzcd} \tag{D.3.12}$$

### Bisimilariry technique

The definition of bisimilarity suggests a proof technique to demonstrate that two states $s$ and $t$ are bisimilar, i.e. find a bisimulation relation containing the pair $(s, t)$, called the bisimulation proof method. The common way of proceeding to prove a bisimilarity $s \sim t$ is starting with a relation $R$ containing at least the pair $(s, t)$ as an initial guess for a bisimulation. We check the bisimulation clauses, if some pairs are missing $R$ is not a bisimulation. We add the pairs, we check the clauses of the new relation to guess for a bisimulation. We repeat the method until a bisimulation is found. Then we search for the smaller bisimulation, this is the bisimulation proof method. A smaller bisimulation, that contains fewer pairs, reduces the amount of bisimulation clauses to check.

The bisimulation proof method has the two interesting features. The first one is the locality of the checks on the states, since we check only the outside transitions. The second one is the no of a hierarchy on the pairs of the bisimulation, since there is no temporal order on the checks is required. As a consequence, bisimilarity can be effectively used to reason about infinite or circular objects. This is in clear contrast with inductive techniques, that require a hierarchy, and that therefore are best suited for reasoning about finite objects. Thus there is a clear contrast with the inductive definitions and the inductive proofs. In the case of induction, there is always a basic case where to start from, followed by an inductive part where one builds on top of what one has so obtained so far. The definition of $\sim$, and its proof technique, are not inductive, but coinductive.

Coinduction is the dual to structural induction, we show the duality with the following schema.

| Induction | Coinduction |
|---|---|
| inductive definitions | coinductive definitions |
| induction technique | coinductive technique |
| constructors | destructors |
| smallest universe | largest universe |
| congruence | bisimulation |
| least fixed-points | greatest fixed-points |

For more details see [3, 46].

### 3.3.3 Bisimilarity as a fixed-point

The main elements of the duality between induction and coinduction are evident. An inductive definition tells us what are the constructors for generating the elements. A coinductive definition tells us what are the destructors for decomposing the elements. The destructors show what we can observe of the elements. If we think of the elements as black boxes, then the destructors tell us what we can do with them. This is clear in the case of infinite lists, and also in the definition of bisimulation. A second element of duality is the definition by means of some rules. If the definition is inductive, we look for the smallest universe in which such rules live. If it is coinductive, we look for the largest universe.

Intuitively a bisimulation is a relation closed under the destructors, thus the dual of bisimulation is a congruence, which is a relation closed under the constructors. Also, we have not explained yet how induction and coinduction are related to least and greatest fixed points. We do this in the next section, where, in fact, we use fixed-point theory to explain the meaning of induction and coinduction.

We recall that the definitions of least and greatest fixed point. On complete lattices generated by the powerset construction, if $F\colon \mathcal{P}(X) \to \mathcal{P}(X)$ is monotone, then

$$\mathrm{lfp}(F) \overset{\text{def}}{=} \bigcap \{\, S \mid F(S) \subseteq S \,\}$$

$$\mathrm{gfp}(F) \overset{\text{def}}{=} \bigcup \{\, S \mid S \subseteq F(S) \,\}\,.$$

*The functional of bisimilarity*

Intuitively, a set A is defined coinductively if it is the greatest solution of an inequation of a certain form. The coinduction proof principle just says that any set that is a solution of the same inequation is contained in A. Dually, a set A is defined inductively if it is the least solution of an inequation of a certain form, and the induction principle then says that any other set that is a solution to the same equation contains A. Familiar inductive definitions and proofs can be formalised in this way. To see how bisimulation and its proof method fit the coinaductive schema, let $(W, A, Tr)$ be an LTS, and consider the function on powersets $F_{\sim} \colon \mathcal{P}(W \times W) \to \mathcal{P}(W \times W)$, so defined.

**Definition 3.33.** *$F_{\sim}$, called the functional associated to bisimulation, is the set of all pairs (s, t) such that:*

1. *for all $s'$ with $s \xrightarrow{a} s'$, there is $t'$ such that $t \xrightarrow{a} t'$ and $s' R t'$*
2. *for all $t'$ with $t \xrightarrow{a} t'$, there is $s'$ such that $s \xrightarrow{a} s'$ and $t' R s'$.*

A simple application of fixed-point theory, in particular the Knaster-Tarski Theorem (that we discuss below), leads to the following theorem.

**Proposition 3.34.** 1. *$\sim$ is the greatest fixed point of $F$*
2. *$\sim$ is the largest relation $R$ such that $R \subseteq F_{\sim}(R)$, thus $R \subseteq \sim$ for all $R$ with $R \subseteq F_{\sim}(R)$.*

We recall that a complete lattice is a partially ordered set where all its subsets have a supremum, called least upper bound. This implies that there are all subsets with an infimum, called greatest lower bound. Using $\leq$ to indicate the partial order, a point x in the lattice is a post-fixed point of an endofunction F on the lattice if $x \leq F(x)$; it is a pre-fixed point if $F(x) \leq x$.

**Theorem 3.35 (Knaster-Tarski).** *Let $(L, \leqslant)$ is a complete lattice and let $f \colon L \to L$ is a continuous function. Then*

1. *$\sqcap \{\, x \in L \mid f(x) \leqslant x \,\}$ is the least fixed point of $f$*
2. *$\sqcup \{\, x \in L \mid x \leqslant f(x) \,\}$ is the greatest fixed point of $f$*

We deduce from the theorem that:

- a monotone endofunction on a complete lattice has a greatest fixed point;
- for an endofunction F on a complete lattice the following rule is sound:

$$\frac{F \text{ monotone} \qquad x \leq F(x)}{x \leq \text{gfp } (F)} \qquad (1)$$

where gfp($F$) indicates the greatest fixed point of F. The existence of the greatest fixed point justifies coinductive definitions, while rule (1) expresses the coinduction proof principle, using Knaster-Tarski theorem.

Proposition 3.34 is a consequence of the Knaster-Tarski theorem because the functional associated to bisimulation gives us precisely the clauses of a bisimulation, and is monotone on the complete lattice of the relations on $W \times W$, in which the join is given by relational union, the meet by relational intersection, and the partial order by relation inclusion:

**Lemma 3.36.**
- *$R$ is a bisimulation iff $R \subseteq F_{\sim}(R)$*
- *$F_{\sim}$ is monotone (that is, if $R \subseteq S$ then also $F_{\sim}(R) \subseteq F_{\sim}(S)$).*

For such functional $F_{\sim}$, the rule (1) asserts that any bisimulation only relates pairs of bisimilar states. Example 3.38 shows that $\sim_\omega$ is not a fixed point for it.

*Approximants of bisimilarity*

We can approximate bisimilarity using the following inductively-defined relations.

**Definition 3.37.** *Let $W$ be the states of an LTS. We define the* approximate bisimilarity *$\sim_\omega$ inductively as*

- $\sim_0 \overset{def}{=} W \times W$
- *$s \sim_{n+1} t$ for $n \geqslant 0$ if*
  - *for all $s'$ with $s \xrightarrow{a} s'$, there is $t'$ such that $t \xrightarrow{a} t'$ and $s' \sim_n t'$*
  - *for all $t'$ with $t \xrightarrow{a} t'$, there is $s'$ such that $s \xrightarrow{a} s'$ and $s' \sim_n t'$*
- $\sim_\omega \overset{def}{=} \bigcap_{n \geqslant 0} \sim_n$.

In general, $\sim_\omega$ does not coincide with $\sim$, as the following example shows.

*Example 3.38.* Suppose $a \in A$, and let $a^0$ be a state with no transitions, $a^\omega$ a state whose only transition is $a^\omega \xrightarrow{a} a^\omega$, and $a^n$, for $n \geqslant 1$, states with only transitions $a^n \xrightarrow{a} a^{n-1}$. Also, let $s, t$ be states with transitions $s \xrightarrow{a} a^n$ for all $n \geqslant 0$ and $t \xrightarrow{a} a^n$ $t \xrightarrow{a} a^\omega$ for all $n \geqslant 0$. By induction on $n$, for all $n$ we can prove that $s \sim_n t$ and hence $s \sim_\omega t$. However, it holds that $s \nsim t$: the transition $t \xrightarrow{a} a^\omega$ can only be matched by $s$ with one of the transitions $s \xrightarrow{a} a^n$. But, for all $n$, we have $a^\omega \nsim a^n$, as only from the former state $n + 1$ transitions are possible. ♮

In order to reach $\sim$, in general we need to replace the $\omega$-iteration that defines $\sim_\omega$ with a transfinite iteration, using the ordinal numbers. However, the situation changes if the LTS is finitely branching, meaning that for all $s$ the set $\{\, s' \mid s \xrightarrow{a} s', \text{ for some } a \,\}$ is finite. In this case, the natural numbers are sufficient.

**Theorem 3.39.** *On finitely branching LTSs, relations $\sim$ and $\sim_\omega$ coincide.*

Also Theorem 3.39, about approximating bisimilarity using the natural numbers, can be seen as an application of fixed-point theory, in which one uses the extra hypothesis of cocontinuity of the functional. Let $\bigcap$ denote the meet operation of the complete lattice; then an endofunction on such a lattice is cocontinuous if for all sequences $\alpha_0, \alpha_1, \ldots$ of decreasing points in the lattice (i.e., $\alpha_i \geqslant \alpha_{i+1}$, for $i \geqslant 0$) we have $F\left(\bigcap_i \alpha_i\right) = \bigcap_i F\left(\alpha_i\right)$.

**Theorem 3.40 (Kleene fixed point).** *For a cocontinuous endofunction $F$ on a complete lattice we have*

$$\mathit{gfp}(F) = \bigcap_{n \geqslant 0} F^n\left(\top\right)$$

*where $\top$ is the top element of the lattice, and $F^n(\top)$ indicates the $n$-th iteration of $F$ on $\top$ $F^0(\top) \overset{\mathit{def}}{=} \top$ and $F^{n+1}(\top) \overset{\mathit{def}}{=} F(F(\top))$.*

The cocontinuity of the functional associated to bisimilarity is guaranteed by the finitely branching property of the LTS, and thus Theorem 3.39 becomes an instance of Theorem 3.40. Without cocontinuity, to reach the greatest fixed point using inductively-defined relations we need to iterate over the transfinite ordinals, as the following theorem shows.

**Theorem 3.41.** *If $F$ is a monotone endofunction on a complete lattice, then there is an ordinal $\alpha$ of cardinality less than or equal to that of the lattice such that for $\beta \geqslant \alpha$ the greatest fixed point of $F$ is $F^\beta(\top)$ where $\top$ is the top element of the lattice and $F^\lambda(\top)$, where $\lambda$ is an ordinal, is so defined $F^0(\top) \overset{\mathit{def}}{=} \top$ and $F^\lambda(\top) \overset{\mathit{def}}{=} F\left(\bigcap_{\beta \geqslant \lambda} F^\beta(\top)\right)$ for $\lambda > 0$.*

As the ordinals are linearly ordered, and each ordinal is either the successor of another ordinal or the least upper bound of all its predecessors, the above definition can also be given thus $F^0(\top) \overset{\text{def}}{=} \top$, $F^{\lambda+1}(\top) \overset{\text{def}}{=} F\left(F^\lambda(\top)\right)$ for $\lambda > 0$ for a successor ordinal, and $F^\lambda(\top) \overset{\text{def}}{=} F\left(\bigcap_{\beta \geqslant \lambda} F^\beta(\top)\right)$ for a limit ordinal. Theorem 3.41 tells us that at some ordinal $\alpha$ the function reaches its greatest fixed point. On ordinals larger than $\alpha$, of course, the function remains on such fixed point. Therefore essentially the theorem tells us that $F^\lambda(\top)$ returns the greatest fixed point of $F$ for all sufficiently large ordinals $\lambda$. In case F is cocontinuous, Theorem 3.40 assures us that we can take $\alpha$ to be the first limit ordinal, $\omega$ (not counting 0 as a limit ordinal). The property dual to cocontinuity, on increasing sequences, least fixed points and joins, is called continuity. Theorems 3.40 and 3.41 give us constructive proofs of the existence of greatest- fixed points. The constructions are indeed at the heart of the algorithms used today for bisimilarity checking. Complete lattices are dualisable structures: we can reverse the partial order and get another complete lattice. Thus the definitions and results above about joins, post-fixed points, greatest fixed points, cocontinuity have a dual in terms of meets, pre-fixed points, least fixed points, and continuity. As the results we gave justify coinductive definitions and the coinductive proof method, so the dual theorems can be used to justify familiar inductive definitions and inductive proofs for sets.

## 3.4 Metric theory

The notions of bisimulation, Section 3.3, are too sensitive to a slight perturbation of the probabilities, that would make two systems non-bisimilar. To solve this problem the approach is to introduce the metrics. Unlike an equivalence relation, a metric can vary smoothly as a function of the probabilities and it can be used to measure the similarity of two systems in a more informative way than an equivalence relation. This motivate a shift to the study of the notion of metric, which provides a tight limit for how much the value of distributions of some actions can differ at states of the system. Given two states $s$ and $t$ the metric distance in the deterministic case can be defined as $\sup_{\varphi \in \Phi} |\varphi(s) - \varphi(t)|$, where $\varphi(\cdot)$ is a distribution on a state. If we focus on bisimilar states, a bisimulation is the kernel of the metric. A standard way to define these metrics is to adapt the characterisation of the bisimulation with fixed point for the metrics. We identify the main rules we are interested of the systems, from these rules we create a functional operator that calculate at each iteration the better distance on a pair of states, the fixed point of the operator is the metric searched.

### 3.4.1 Metric instead of logic

In literature [7] probabilistic bisimulation is commonly characterised using a negation free logic $\mathcal{L}$: $T|\varphi_1 \wedge \varphi_2|\langle a\rangle_q\varphi$, where $a$ is a label from the set of labels $L$ and $q \in [0, 1)$ is a rational number. It is relevant that two plMcs are bisimilar if and only if their start states satisfy the same formulas, i.e. given the states $t, s$ and the formula $\varphi$, then $t \models_P \varphi$ and $s \models_P \varphi$ where $P$ is the plMc. There exist several alternate characterisations of probabilistic bisimulation, that use functions into $\mathbb{R}$ instead of the logic $\mathcal{L}$. We define a set of functions which are sufficient to characterise bisimulation. We give an explicit syntax of a set of functional expressions, which become functions when we interpret them in a system. Thus when we move from one system to another we have the same functional expression, nevertheless we may say "the same function". This is no different from having syntactically defined formulas of some logic which become boolean-valued functions when they are interpreted on a structure. We now give the class of functional expressions. First, we give some notation. Let $\lfloor r \rfloor_q = r - q$ if $r > q$, and 0 otherwise. $\lceil r \rceil_q = q$ if $r > q$, and r otherwise. Note that $\lfloor r \rfloor_q + \lceil r \rceil_q = r$. For each $c \in (0, 1]$, we consider a family $F_c$ of functional expressions generated by the following grammar. Here q is a rational in $[0, 1]$.

$$\begin{array}{rll} f^c ::= & \lambda s.1 & \text{Constant schema} \\ \mid & \lambda s.1 - f^c(s) & \text{Negation schema} \\ \mid & \lambda s.\min(f_1^c(s), f_2^c(s)) & \text{Min schema} \\ \mid & \lambda s.\sup_{i \in \mathbb{N}}\{f_i^c(s)\} & \text{Sup schema} \\ \mid & \lambda s.c\int_{t\in S} \tau_a(s,t) f^c(t) & \text{Prefix schema} \\ \mid & \lambda s.\lfloor f^c(s) \rfloor_q & \\ \mid & \lambda s.\lceil f^c(s) \rceil_q & \text{Conditional schemas} \end{array}$$

$F_+^c$ is the sub-collection of $F_c$ that does not use the negation schema.

*Property 3.42.* The functions $1, 1 - f, \min(f_1, f_2), \lfloor f \rfloor_q, \lceil f \rceil_q$ can be used to approximate any continuous Lipschitz function from [0, 1] to [0, 1].

This shows that we can replace the constant schema, negation schema and conditional schemas with one schema: $\lambda s.g(f(s))$, where $g$ is any continuous Lipschitz function. To get positive functions $F_+^c$, we can restrict $g$ to monotone continuous Lipschitz functions. A routine induction on the structure of the functional expression $f^c \in F_+^c$, shows the monotone property.

**Lemma 3.43 (monotone).** *If $P$ is a sub-plMc of $Q$, then $(\forall f \in F_+^c)(\forall s \in S_P)[f_P^c(s) \leq f_Q^c(s)]$.*

Each function of a class $F$ assigns a value in the interval $[0, 1]$ to states of a plMc. The result of evaluating these functions at a state corresponds to a quantitative measure of the extent to which the state satisfies a formula of L. The identification of this class motivates the intuition of using a metric $d$, which is a function that yields a real number distance for each pair of processes. It should satisfy the usual metric conditions, i.e. $d(P, Q) = 0$ implies $P$ is bisimilar to $Q$, $d(P, Q) = d(Q, P)$ and $d(P, R) \leqslant d(P, Q) + d(Q, R)$. We formalise this intuitions defining a family of metrics $\{\, d_c \mid c \in (0, 1] \,\}$. These metrics support the spectrum of possibilities of relative weighting of the two factors that contribute to the distance between processes. The complexity of the functions distinguishing them versus the amount by which each function distinguishes them. $d_1$ captures only the differences in the probability numbers, probability differences at the first transition are treated on par with probability differences that arise very deep in the evolution of the process. In contrast, $d_c$ for $c < 1$ give more weight to the probability differences that arise earlier in the evolution of the process, i.e. differences identified by simpler functions. As $c$ approaches 0, the future gets discounted more. As is usual with metrics, the actual numerical values of the metric are less important than properties like the significance of zero distance, relative distance of processes, contractivity and the notion of convergence.

Each collection of functional expression $F_c$ be the set of all such expressions induces a distance function as follows: $d^c(P, Q) = \sup_{f^c \in F^c} |f_P^c(s_P) - f_Q^c(s_Q)|$.

**Theorem 3.44.** *For all $c \in (0, 1]$, $d^c$ is a metric.*

Each of these metrics agree with bisimulation, indeed $d_c(P,Q)=0$, if and only if $P$ and $Q$ are bisimilar. For $c<1$, we show how to compute $d_c(P,Q)$ to within $\varepsilon$. To show this we need the relation between plMcs and formulas. For any state in a finite plMc that satisfies a formula, there is a partial witness from $F^c_+$. Given any $\varphi \in \mathcal{L}$ and a finite plMc P, and any $c \in (0,1]$, there is a functional expression $f^c \in F^c_+$ such that 1) $\forall s \in S_P . f^c_P(s) > 0$ iff $s \models_P \varphi$, and 2) for any plMc Q for all $s \in S_Q . s \not\models_P \varphi \implies f^c_Q(s) = 0$. In [cita articolo Metrics for labeled markov systems] the authors have showed the following theorem.

**Theorem 3.45.** *For any plMc P,* $(\forall c \in (01]), \forall s, s' \in S_P$ *then* $\Big((\forall \varphi \in \mathcal{L})\, s \vDash_P \varphi \longleftrightarrow s' \vDash_P \varphi\Big) \longleftrightarrow (\forall f \in F^c)[f^c_P(s) = f^c_P(s')]$.

*Example 3.46.* Consider the plMc $P$ with two states, and a transition going from the start state to the other state with probability $p$. Let $Q$ be a similar process, with the probability $q$. Then we show that $d_c(P,Q) = c|p-q|$. Now if we consider $P'$ with a new start state, which makes a $b$ transition to $P$ with probability 1, and similarly $Q'$ whose start state transitions to $Q$ on $b$ with probability 1, then $d_c(P',Q') = c^2|p-q|$, showing that the next step is discounted by $c$. ♮

*Example 3.47.* Consider the family of plMcs $\{\, P_\varepsilon \mid 0 \leqslant \varepsilon < r \,\}$ where $P_\varepsilon = a_{r-\varepsilon}.\ Q$, i.e. $P_\varepsilon$ is the plMc that does an $a$ with probability $r-\varepsilon$ and then behaves like $Q$. We demand that $d(P_{\varepsilon_1}, P_{\varepsilon_2}) \leqslant |\varepsilon_1 - \varepsilon_2|$. This implies that $P_\varepsilon$ converges to $P_0$ as $\varepsilon$ tends to 0. We evaluate the function expression $(\langle a \rangle .1)^c$ to $r - \varepsilon c$ at $P_\varepsilon$. This functional expression witnesses the distance between any two $P$ (other functions will give smaller distances). Thus, we get $d(P_{\varepsilon_1}, P_{\varepsilon_2}) = c|\varepsilon_1 - \varepsilon_2|$. ♮

### 3.4.2 Kantorovich metric

A metric provides a way of measuring the distance between two distributions. A famous and useful metric is the Kantorovich metric, that has a natural interpretation in terms of the transportation problem. More information in [9, 15, 27]. We describe the problem in the following. Let $(\Omega, d)$ is a metric space and $\nu, \nu'$ are two measures of total mass 1, i.e. $\sum_{s\in\Omega} \nu(s) = 1$ and $\sum_{s\in\Omega} \nu'(s) = 1$. These measure $\nu$ and $\nu'$ are two divisions of the masses on $\Omega$. The problem of Monge-Kantorovich consists of founding the minimal transportation from a mass described by $\nu$ to a division $\nu'$. The cost of moving a mass $k$ from a location $x$ to location $y$ is $k \cdot d(x,y)$, then we describe a transport by a measure $\nu^2$ on $\Omega \times \Omega$. When $\nu^2$ is simple, the probability of $(x,y)$ represents the mass moved from point $x$ to point $y$. The cost of this transport is $\int_{(x,y)\in\Omega\times\Omega} d(x,y)\, d\nu^2$, with canonical projections $\pi_x, \pi_y \colon \Omega \times \Omega \to \Omega$. We insert the constraint that $\nu^2$ is a transport from $\nu$ to $\nu'$, that means to impose the marginals $\pi_x(\nu^2) = \nu$ and $\pi_y(\nu^2) = \nu'$. The Kantorovich-Rubinstein theorem declares that if $\Omega$ is complete and separable, then there exists an optimal transport with minimal cost equal to the Hutchinson distance between $\nu$ and $\nu'$. We give the definition of Hutchinson metric.

**Definition 3.48.** *Let $\Omega$ is a space, the Hutchinson distance between two measures $\mu$ and $\nu$ is defined by*

$$d_H(\mu, \nu) = \sup_{f \in \langle \Omega \to \mathbb{R}^+ \rangle_1} \left| \int_{x \in \Omega} f(x) d\mu - \int_{x \in \Omega} f(x) d\nu \right|,$$

*where $\langle \Omega \to \mathbb{R}^+ \rangle_1$ is the set of all the measurable functions 1-Lipschitz $f \colon \Omega \to \mathbb{R}^+$.*

The Kantorovich metric $d(\mu, \nu)$ gives an alternative characterisation of the problem via the Borel probability measures, which we introduced in Def. 3.7. We denote by $B_m(\Omega)$ the set of all Borel probability measures on $\Omega$, as consequence for all $z \in \Omega$, if $\mu \in B_m(\Omega)$ then $\int_\Omega d(x, z)\mu(x) < \infty$. We write $M(\mu, \nu)$ for the set of all Borel probability measures on the product space $\Omega \times \Omega$ with marginal measures $\mu$ and $\nu$, i.e. if $m \in M(\mu, \nu)$ then $\int_{y \in \Omega} dm(x, y) = \mu(x)$ and $\int_{x \in \Omega} dm(x, y) = \nu(y)$ hold.

**Definition 3.49.** *For $\mu, \nu \in B_m(\Omega)$, we define the Kantorovich metric $d_K$ as*

$$d_K(\mu, \nu) = \inf \left\{ \int_{\Omega \times \Omega} d(x, y)\, dm(x, y) \;\middle|\; m \in M(\mu, \nu) \right\},$$

*where $M(\mu, \nu)$ is the set of all the joint probability distribution measure with marginals $\mu$ and $\nu$, $d \colon \Omega \times \Omega \to \mathbb{R}$ is a given distance.*

When the distance is not explicit, we refer to the distance $d_R \colon \Omega \times \Omega \to \mathbb{R}$ defined by

$$d_R(s, s') = \begin{cases} 0 & \text{if } s \, R \, s' \\ 1 & \text{otherwise.} \end{cases} \tag{3.13}$$

The function $m$, defined on the set of joint distributions with marginal distributions $\mu(s), \nu(t)$ of the Kantorovich metric, sounds like a weighting function $w \colon \Omega \times \Omega \to \mathbb{R}$ (Def. 3.24). At a closer look this metric is a lifting function, since it lifts a given distance $d \colon \Omega \times \Omega \to \mathbb{R}$ in $d_K \colon \mathrm{Dist}(\Omega) \times \mathrm{Dist}(\Omega) \to \mathbb{R}$. This property have been highlighted by Van Breugel and Worrell [63, 65].

Many problems in computer science only involve finite state spaces, so discrete distributions with finite supports are sometimes more interesting than continuous distributions. For two discrete distributions $\mu$ and $\nu$ with finite supports $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_m\}$, respectively, minimising the total cost of a discretised version of the transportation problem reduces to the following linear programming problem

$$d_{MK}(\mu, \nu) = \min \sum_{x \in Q} \sum_{y \in Q} d(x, y) \cdot m(x, y) \tag{3.14}$$

$$\begin{aligned} \text{conditions: } & \forall x \in Q \,.\, \sum_{y \in Q} m(x,y) = \mu(x) \\ & \forall y \in Q \,.\, \sum_{x \in Q} m(x,y) = \nu(y) \\ & \forall x, y \in Q \,.\, m(x,y) \geqslant 0 \end{aligned} \tag{3.15}$$

### 3.4.3 Metric and bisimulation

[12] States close in the metric should yield similar values. Specifically, for any game structure $G$ and states $s, t$ of $G$ a bisimulation metric needs the continuity property on the pair $s, t$. This definition of bisimulation require, for every mixed move from s, the existence of a mixed move from t, such that the moves induce probability distributions over successor states that are equivalent modulo the underlying bisimulation. Unfortunately, the generalisation of this definition to games fails. Thus the definition has been phrased in terms of expectations of certain metric-bounded quantities.

First we give the definition of metrics for Markov decision processes on probabilistic distributions, which is the traditional definition and called *a posteriori* form. Then we transform this definition in a metric on expectation values, called *a priori* form. The latter form satisfies the reciprocity property, i.e. the probability that player 1 achieves a goal $\psi$ is one minus the probability that player 2 achieves the goal $\neg\psi$ (i.e. not $\psi$). Reciprocity ensures that there is one, canonical, notion of game equivalence. Neither the logical characterization nor the reciprocity result hold for the a posteriori metrics and relations.

#### Probabilistic bisimulation on MDP as fixpoint

In Section 3.3.3 we have shown how to generate a functional associated to a bisimulation, the purpose was using this functional to generate an algorithm to compute if two states are bisimilar or not. In this section the purpose is the converse, we construct a bisimulation by the fixed point of a functional, called $H$, that we will produce.

We define the probabilistic bisimulation (3.31) on MDPs as a fipoint of a new functional/relation called $F$, which maps probabilistic distributions on states. We generate $F$ by lifting any given relation between states. To give an idea considering the colours the elements of a space and the function the distance between them. The basic distance measures dissimilarity between individual colours. If we consider image colours distributions, then a dissimilarity is a lifted distance. We have recall the lifting definition in Def. 3.24 and here we use the notation $\varDelta$. Given a MDP $M$, it is also a game structure and thus $[v](s)$ is the interpretation of the variable $v$ on a state $s$ with real value between $[0,1]$. First we define the relation $[\equiv] \in M$ for all $s, t \in S$ as

$$[s \equiv t] = \max_{v \in V} |[v](s) - [v](t)|.$$

**Definition 3.50 (Relation transformer).** *The* relation transformer $F\colon 2^{S\times S} \to 2^{S\times S}$ *is defined, for all states* $s, t \in S$ *and for all relations* $R \subseteq S \times S$*, as*

$$s \equiv t \ \wedge \ \forall x_1 \in D_1(s) . \exists y_1 \in D_1(t) \ . \ \delta(s, x_1) \sqsubseteq_R \delta(t, y_1).$$

We define the fixpoint of $F$, this corresponds to bisimulation on MDPs.

**Definition 3.51 (Probabilistic (bi)simulation on MDPs).** *The* probabilistic simulation *is the greatest fixpoint of F. The* probabilistic bisimulation *is the greatest symmetrical fixpoint of F.*

### Metric for a probabilistic simulation

We relax the strictly definition of probabilistic bisimulation by introducing the definition of metric. To obtain a metric we lift the fixpoint of relation transformers 3.50, i.e. from subsets of $S^2$ to maps $S^2 \to \mathbb{R}$. Before defining the new transformer, we introduce a distance of distributions that we will define later. The *distribution distance* $D(\mu, \nu)(d)$, with $\mu, \nu \in Dist(S)$ and for all metric $d \in M$, is a measure of the cost $d(s, t)$ of moving a unit of probability mass form $s \in S$ to $t \in S$.

**Definition 3.52 (Metric transformer).** *For* $s, t \in S$ *and for all metric* $d \in M$*, a* metric transformer $H^{\scriptscriptstyle MDP}_{\scriptscriptstyle post}\colon M \to M$ *is defined as*

$$H^{\scriptscriptstyle MDP}_{\scriptscriptstyle post}(d)(s, t) = [s \equiv t] \sqcup \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} D(\delta(s, x_1), \delta(t, y_1))(d), \tag{3.16}$$

*where* $a \sqcup b = \max\{a, b\}$.

The distribution distance $D(\mu, \nu)(d)$ is the metric between $\mu, \nu$, it is defined via the trans-shipping problem, as the minimum cost of shipping the distribution $\mu$ into $\nu$, with edge costs d. Thus, $D(\mu, \nu)(d)$ is the solution of the following linear programming problem over the set of variables $\{\lambda_{s,t}\}_{s,t \in S}$

$$\text{Minimise } \sum_{s,t \in S} d(s, t) \lambda_{s,t}$$

$$\text{subject to } \sum_{t \in S} \lambda_{s,t} = \mu(s), \sum_{s \in S} \lambda_{s,t} = \nu(t), \lambda_{s,t} \geqslant 0.$$

**Definition 3.53.** *The* simulation metric *in MDPs is defined as the least fixpoint of 3.16, since equivalent states should have distance 0. The* bisimulation metric *is defined as the least symmetrical fixpoint of 3.16.*

Every solution for a linear programming problem gives a bound on the optimal value of the objective function. The problem can be converted into a dual problem. We define in the following the dual problem of $D(\mu,\nu)(d)$ with respect to a metric $d \in M$ and $C(d) \subseteq \mathcal{F}$ be the subset of valuations $k \in \mathcal{F}$ such that $k(s) - k(t) \leqslant d(s,t)$ for all $s, t \in S$. Then the dual formulation is the following.

$$\text{Maximise } \sum_{s\in S} \mu(s)k(s) - \sum_{s\in S} \nu(s)k(s) \quad \text{subjects to } k \in C(d). \tag{3.17}$$

We replace the definition of distance 3.17 into the equation 3.16

$$H^{\text{MDP}}_{\text{post}}(d)(s,t) = [s \equiv t] \sqcup \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} \sup_{k \in C(d)} \left(\mathbb{E}^{x_1}_s(k) - \mathbb{E}^{y_1}_t(k)\right). \tag{3.18}$$

We call the metric transformer $H^{\text{MDP}}_{\text{post}}$ the a posteriori metric transformer, since the valuation $k$ in equation 3.18 is chosen after the moves $x_1$ and $y_1$ are chosen. We can define an a priori metric transformer, where $k$ is chosen before $x_1$ and $y_1$.

$$H^{\text{MDP}}_{\text{prio}}(d)(s,t) = [s \equiv t] \sqcup \sup_{k \in C(d)} \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} \left(\mathbb{E}^{x_1}_s(k) - \mathbb{E}^{y_1}_t(k)\right). \tag{3.19}$$

Given a MDP $M$ we calculate the distance between two states $a, b \in M$, the results of a posteriori metric and a priori metric coincide, i.e. $H^{\text{MDP}}_{\text{post}} = H^{\text{MDP}}_{\text{prio}}$ as shown in Theorem 3.1 in [12].

**Definition 3.54 (bisimulation).** *$[\sim] \in M$ is the least symmetrical fixpoint of $H^{MDP}_{post}$ of a MDP $M$.*

Consider a game structure with only a single player is not useful for our purpose, if we want to redefine the bisimulation with probability we have to consider a game with two independent players. the a priori and the a posteriori metrics do not coincide over games. A posteriori metrics are defined via the metric transformer $H_{\sqsubseteq_1} : M \to M$ as follows, for all $d \in M$ and $s, t \in S$.

$$\begin{aligned} H_{\sqsubseteq_1}(d)(s,t) &= [s \equiv t] \sqcup \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} \sup_{y_2 \in D_1(s)} \inf_{x_2 \in D_1(t)} D(\delta(s,x_1,x_2), \delta(t,y_1,y_2))(d) \\ &= [s \equiv t] \sqcup \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} \sup_{y_2 \in D_1(s)} \inf_{x_2 \in D_1(t)} \sup_{k \in C(d)} \left(\mathbb{E}^{x_1\,x_2}_s(k) - \mathbb{E}^{y_1\,y_2}_t(k)\right) \end{aligned} \tag{3.20}$$

A priori metrics are defined by bringing the $\sup_k$ outside. Precisely, we define a metric transformer $H_{\preccurlyeq_1} : M \to M$ as follows, for all $d \in M$ and $s, t \in S$.

$$\begin{aligned} H_{\preccurlyeq_1}(d)(s,t) &= [s \equiv t] \sqcup \sup_{k \in C(d)} \sup_{x_1 \in D_1(s)} \inf_{y_1 \in D_1(t)} \sup_{y_2 \in D_1(s)} \inf_{x_2 \in D_1(t)} \left(\mathbb{E}^{x_1\,x_2}_s(k) - \mathbb{E}^{y_1\,y_2}_t(k)\right) \\ &= [s \equiv t] \sqcup \sup_{k \in C(d)} \left( \sup_{x_1 \in D_1(s)} \inf_{x_2 \in D_1(t)} \mathbb{E}^{x_1\,x_2}_s(k) - \sup_{y_1 \in D_1(s)} \inf_{y_2 \in D_1(t)} \mathbb{E}^{y_1\,y_2}_t(k) \right) \\ &= [s \equiv t] \sqcup \sup_{k \in C(d)} \left(Pre_1(k)(s) - Pre_1(k)(t)\right). \end{aligned} \tag{3.21}$$

**Lemma 3.55 (Monotonic).** *The functions* $H_{\preccurlyeq_1}$ *and* $H_{\sqsubseteq_1}$ *are monotonic in the lattice of metrics* $(M, \leqslant)$*.*

On the basis of this lemma, we can define the least fixpoints of $H_{\preccurlyeq_1}$ and $H_{\sqsubseteq_1}$, which will yield our game simulation and bisimulation metrics.

**Definition 3.56.** *The a priori simulation metric* $[\preccurlyeq_1]$ *is the least fixpoint of* $H_{\preccurlyeq_1}$*. The a priori bisimulation metric* $[\simeq_1]$ *is the least symmetrical fixpoint of* $H_{\preccurlyeq_1}$*.*
*The a posteriori game simulation metric* $[\sqsubseteq_1]$ *is the least fixpoint of* $H_{\sqsubseteq_1}$*. The a posteriori game bisimulation metric* $[\cong_1]$ *is the least symmetrical fixpoint of* $H_{\sqsubseteq_1}$*.*

# 4

# Security analysis with $\varepsilon$-simulations

In this chapter we consider security analysis of concurrent systems with probability elements. Describing the systems with the probabilistic automata we use the notion of bisimulation and simulation to check the security constraints. We recall the notions of polynomially accurancy of security constraints by recalling the $\varepsilon$-simulations. In Section 4.1 we introduce security and cryptographic and list their importance in the real systems. In Section 4.2 we describe the systems as a probabilistic automata. In Section 4.3 we introduce the $\varepsilon$-simulations.

## 4.1 Security and cryptography

Security and cryptography are branches of computer science that have an immediate relevance and feedback in the present world. A computer program can execute a procedure in another computer on a shared network, commonly Internet. Internet is the global system of interconnected computer networks and gives the possibility of using remote procedures between wherever computers. The remote interaction causes the problem of security identity and security messages, which is a central problem in the present society. As example, we can think to email exchange and login in a bank account. With Internet diffusion the network protocols has been refined introducing a presection with cryptographic methods in the algorithms, called cryptographical protocols and encryption protocols. These security protocols are considered safe only when we are able to check that each adversary is not able to attack it successfully. The set of the possible attacks is called the problem of cryptographic protocol verification, it has been largely studied in literature. To be sure that at least the theoretical protocol is correct, the better way is to perform the analysis using a formal and rigorous model that ensures the correctness of our reasoning. Thus, following [62], we focus on Probabilistic Automata (PA) model, which provides the mathematical rigour that is necessary to study rigorously randomised systems. This model permits protocol verification approaches, where messages are represented as symbols and the adversary can try to attack the protocol using a restricted set of actions, which usually does not include the possibility to guess secrets as private keys. This means that PA model provides the tools to relate executions of different systems, simulations and bisimulations allow us to compare the computations of two systems and to say if they behave in the same way or if their behaviours are not similar.

To reduce the possibility of an attack, and also the chance that an intruder can obtain reserved informations, we can use random values and randomised primitives. As example we consider the nonce, it is a random number used once during the authentication protocol to ensure that old communications cannot be reused. But this implies that, when we analyse the correctness of a protocol, we must consider also all probabilistic aspects that occur in the protocol. Furthermore another source of difficulty when we generate the ideal model is the nondeterminism, induced by several components that interact with each other. These components have different execution speeds and possible input from users, for example different protocol participants, adversary, external entities (as a key generator). Moreover, we can use nondeterminism to model underspecification and abstraction. We use the first one when we are not interested to specify all details of a component, we have only need to know the actions available and not all the details. The second one when we derive no desired properties from the knowledge of an element, then its actual probability values is ignored. The analysis of systems that present both probability and nondeterminism is a problem already studied in the context of randomized distributed algorithms. In particular, we can find several common aspect between cryptography and distributed algorithms. For example, in both cases we have to consider probabilistic and nondeterministic behaviors and the analysis is usually performed comparing the executions of two systems that behave in a similar way or transforming the execution of one system to the execution of another one that represents an attacker. One example of the last case is the analysis of indistinguishability property: given an attacker, we build another machine which behaviour is very close to the one of the original attacker and that it is able to break the indistinguishability property.

Concurrency theory allows us to prove properties of randomised distributed algorithm in an hierarchical and compositional way. The possibility to work hierarchically permits to model the problem at several level of abstraction, and each level represents the algorithm in a more or less detailed way. This means, for example, that we can define an abstract level where almost all probabilistic aspects are missing and where we can focus our attention on the specification of the problem, studying its properties to see if the specification satisfies our requirements. If it is the case, then we can define other levels where we detail the single components that form the overall algorithm. The compositionality allows us to study each single component independently from the others and to extends its properties to the overall system. This means that if we prove that a component at level $i$ is an implementation of the same component in the level $i+1$, then we can say that the overall system at level $i$ is an implementation of the system at level $i+1$. So this implies that whenever we generate a chain of implementations from the fully detailed system to the abstract system, by transitivity we can conclude that the fully detailed system is an implementation of the abstract system and thus it satisfies the same properties of the specification. The compositionality is very useful since it permits to consider small independent components instead of a big system with several interacting modules and moreover we can reuse already known results in several proofs. So, if we already know that a functionality is implemented by a specific component, then we simply replace the functionality with it and by compositionality we derive that the new system is an implementation of the old one without proving it another time.

## 4.2 Polynomially accurate simulation relation

The main element of the compositionality is the implementation, which is transitive and it has logical characterization that permits to know which kind of properties the implementation relation preserves. But it is difficult to check this property in a system. Fortunately, the simulation and bisimulation relations allow us to derive global properties of objects checking the properties preserved by each computational step of the system independently from the other steps. In fact, usually we reason about the properties that are satisfied in a state and the ones that are satisfied after performing an execution step. Then, just composing all reasoning we derive the properties that are valid in the global object. Since probabilistic automata are an useful framework that help the analysis of randomized distributed system, we want to check if the probabilistic automata can help the verification of cryptographic protocols. To do this, we use the probabilistic automata directly in the computational model and to utilise the mathematical rigour and the simulation relations of the framework to study the correctness of cryptographic protocols.

The use of an ideal model generates a problem. A real adversary can always generate a message that breaks the protocol, he can generate a random sequence of bits that violates the protocol. In the ideal model the adversary can not generate a message randomly, but it can only derive new messages from old ones. This implies that standard simulation relations of probabilistic automata model cannot be used to relate real and ideal adversaries, because a real attacker can generate messages and hence it can perform actions that an ideal adversary can not simulate. For this reason Turrini ( [62]) defines an extension of simulation relations, which permits to match the step condition up to some error. Moreover, adding an error in the simulations the computational model verifies that both the security of the protocol and the computational power of the adversary depend on a security parameter.

In the computational model both agents and adversaries are parameterized by a security parameter. This means that for each value of security parameter, we have different adversaries and agents that work always in the same way but using different values (for example, the length of nonces) and usually the security parameter is used to fix the computational power of adversaries. Hence we should extend the simulation relation to consider also families of automata (and not only single automata), families that are parameterized by the security parameter. To consider the computational power of adversary, we can extend ordinary simulations adding some information about how many steps we have spent to reach a particular state of the automaton. To do this, we can base our simulation on automaton executions instead of automaton states, because an execution describes the sequence of states and actions that has led to its final state. In this way we are able to provide an upperbound to the computational power of an adversary bounding the execution lengths, that can be related to the security parameter via a polynomial, for example. As we said previously, a real adversary can generate messages that an ideal adversary can not simulate but such messages must have negligible probability (otherwise the protocol is not secure). To consider these messages, we can extend simulations permitting that the matching transition matches up to an error. If we relate such error to the security parameter, then we can provide an upperbound to the global error made by a real adversary with respect to an ideal adversary after a given number of steps. In this way, if we force the number of steps to be polynomial with respect to the security parameter and the step error to be negligible, then the global error is negligible and hence the protocol is secure with respect to the computational model meaning.

Main advantages of this simulation are that we can fix an upperbound to the adversarial adversary bounding the length of executions; we can decide if the probability of unmatched executions is negligible and hence to decide if there exist attacks such that their probabilities of success are not negligible. Moreover, the verification of the protocol correctness is local, step-based and not global. In this way we can focalize our attention to a restricted set of adversarial actions and this permits to simplify the verification. This holds because the check of the step condition reduces directly to the statement of correctness of the underlying cryptographic protocols and if we have considered enough level of abstractions, for each simulation we can analyze a single cryptographic aspect: a simulation considers only that nonces are not repeated, another considers only signs, and so on. On the contrary, when the analysis is global, we must consider all computational aspects at the same time and this makes the correctness proof more difficult.

This new simulation becomes useful if we can prove some properties of it. An indispensable result we need is the one that allows us to extends results on single steps to the complete chain of steps because if we do not have such result, then it could not be easily used in a hierarchical correctness proof. In fact, in the hierarchical approach we define several levels of abstraction, we prove a simulation from a level to the next one and then we obtain a simulation from the lowest to the highest level of abstraction. Without such result, we can not use intermediate result to relate the lowest level to the higher, but we must prove the simulation directly but this proof could be not so easy to obtain. We can obtain such result proving the transitivity or other properties of the polynomially accurate simulations. Another useful property is the composition property, that permits to preserve the simulation when we put other automata (like agents or other adversaries) beside simulation related automata. So, for example, if we have that A is simulated by B, then for each C we have that A composed C is simulated by B composed C. Once we have the compositional property, we can model real and ideal cryptographic primitives using automata and then relate them using our new simulation. In this way we obtain a library of basic results that can be used each time we want to replace a real primitive with its ideal counterpart. This library allows us to easily prove the existence of the simulation between contiguous levels defined in the hierarchical correctness proof. We can also check if the new simulation is a conservative extension of literature simulations. In this case, ordinary simulations are particular cases of our relation and the study of the logical characterization or of relations with other models is probably easier.

The idea described is a way to relate the ideal and the computational model: we base our modeling on probabilistic automata and we represent each actor (cryptographic primitive, protocol, adversary, and so on) with an automaton. In particular, we model them as in the computational model: exchanged messages are bitstrings which length depends on a security parameter k that parameterize the automaton; the probabilistic aspects are considered directly by the transitions of the involved automata. Once we have the automata that model all actors, their composition models the concrete protocol that interacts with the concrete adversary. The correctness proof is obtained relating this concrete automaton with another automaton that implements actors ideally. This means that in this abstract automaton, encryptions satisfy the properties as in the ideal model, the adversary can generate a message m only if it is able to derive m from its knowledge, and so on. Standard relations defined on probabilistic automata are not suitable for our purpose, since they do not consider the computational constraints we impose to the adversary. For this reason, we have defined a new simulation relation that takes account of the length of the execution but it is still too restrictive for our aims. In fact, we have that in the ideal model, the probability to decrypt an encrypted message is zero if we do not know the decryption key; in the computational model, the same event has negligible probability. So we can not use an exact matching, but we need to match up to an error. This consideration leads us to define the polynomially accurate simulation that takes account of the security parameter that characterizes the concrete primitives, the computational aspects of the system and the admitted error.

aim define a relation where transitions are matched up to some error that is smaller than any polynomial in some security parameter k provided that computations are of polynomial length. First step: define a relation that can see lengths of computation. For this purpose, we define a relation on sets of executions rather than sets of states. This definition is based on a derived notion of transition that shows how finite executions evolve in a single step.

**Definition 4.1.** *We say that there is a step from a finite execution* $\alpha$ *to a measure* $\nu \in Disc(Execs(A))$*, denoted by* $\alpha \to \nu$*, if there exists a transition* $(lstate(\alpha), a, \mu)$ *such that, for each finite execution* $\alpha a s, \nu(\alpha a s) = \mu(s)$*.*

Now we are able to define a simulation that relates executions instead of single states. This allows us to know how many steps we have performed in a computation, since we can obtain them from the length of the execution.

**Definition 4.2.** *An execution simulation from a probabilistic automaton* $A_1$ *to a probabilistic automaton* $A_2$ *is a relation* $R$ *from* $Execs(A_1)$ *to* $Execs(A_2)$ *such that:*

- $\bar{s_1} \, R \, \bar{s_2}$
- *for each pair* $\alpha_1 \, R \, \alpha_2$*, if* $\alpha_1 \to \nu 1$*, then there exists* $\nu_2$ *such that* $\alpha_2 \to \nu_2$ *and* $\nu_1 \, L(R) \, \nu_2$*.*

The introduction of errors in execution simulations is then straightforward.

**Definition 4.3.** *An* $\varepsilon$*-simulation from a probabilistic automaton* $A_1$ *to a probabilistic automaton* $A_2$ *is a relation* $R$ *from* $Execs(A_1)$ *to* $Execs(A_2)$ *such that:*

- $\bar{s_1} \, R \, \bar{s_2}$
- *for each pair* $\alpha_1 \, R \, \alpha_2$*, if* $\alpha_1 \to \nu 1$*, then there exists* $\nu_2$ *such that* $\alpha_2 \to \nu_2$ *and* $\nu_1 \, L(R, \varepsilon) \, \nu_2$*.*

## 4.3 $\varepsilon$-(bi)simulation

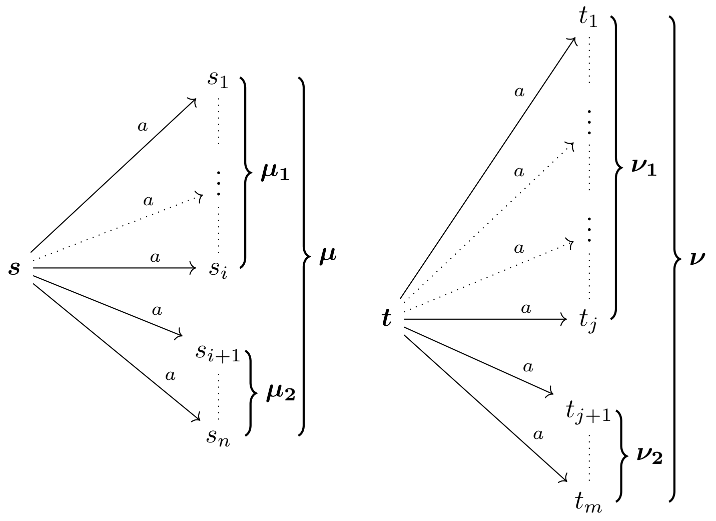
In literature two states are bisimilar only if the probabilistic distributions of outgoing transitions match exactly, where the distributions are the basis of the mathematical models for nondeterministic probabilistic systems. This leads to discard from the bisimulation also the processes that have closed behaviours, since their distributions are closed but not identical. This definition of bisimulation is too exact by using probabilistic distributions, since it is not robust with respect to small variation of the transition probabilities.

In this section we recall the notion of $\varepsilon$-bisimulation, that is a lifting relation from bisimulation on states to an approximate bisimulation on distributions of these states. In the first step we extend the notion of lifting inserting the error $\varepsilon$, we give also some properties. In the second step we give the definition of approximate (bi)simulation.

### 4.3.1 $\varepsilon$-lifting

Now we recall the relation of $\varepsilon$-(bi)simulation and define the $\varepsilon$-relation via the notion of $\varepsilon$-lifting, whose pairs of states have distributions which are not related exactly, but up to some error $\varepsilon$.

We relax the notion of lifting, Def. 3.24, inserting an error that we refer with the notation $\varepsilon$. This signifies that a distribution can be decomposed into two parts, where the second is related to the error allowed and we call "error part". Given a state $s$ and a transition $s \xrightarrow{a} \mu$ we decompose $\mu$ into $\mu_1$ and $\mu_2$, as represented in the left hand side of Diagram 4.1.

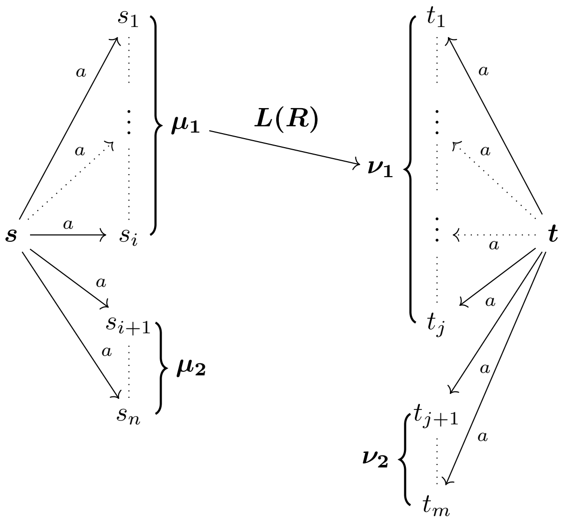


(D.4.1)

For a second distribution $\nu$, resulting from the transition $t \xrightarrow{a} \nu$ represented in the right hand part of the diagram, we search a decomposition with the same error $\varepsilon$ such that $\mu_1$ and $\nu_1$ are in lifting-$R$ relation as represented in Diagram 4.2. Thus the error part can be ignored.

**Definition 4.4 ($\varepsilon$-lifting).** *Let $R \subseteq X \times Y$ be a relation and let $\varepsilon \geqslant 0$. The $\varepsilon$-*lifting *of $R$ is a relation $L(R, \varepsilon) \subseteq Disc(X) \times Disc(Y)$ defined as follows. For each pair $\mu$ and $\nu$ of probability measures on $X$ and $Y$, respectively,*

- $\varepsilon \geqslant 1 \implies \mu\, L(R, \varepsilon)\, \nu$
- $\varepsilon \in [0, 1) \implies \mu L(R, \varepsilon) \nu$ *if there exists $\mu_1, \mu_2 \in Disc(X)$ and $\nu_1, \nu_2 \in Disc(Y)$ such that*
  - $\mu = (1 - \varepsilon)\mu_1 + \varepsilon\mu_2$
  - $\nu = (1 - \varepsilon)\nu_1 + \varepsilon\nu_2$
  - $\mu_1\, L(R)\, \nu_1$.



(D.4.2)

**Proposition 4.5.** *For each relation $R \subseteq Q_1 \times Q_2$ and $\varepsilon = 0$ we have $L(R, 0) = L(R)$.*

*Proof.*

(⇒) Let $\mu_1 \in \mathrm{Disc}(Q_1), \mu_2\, \mathrm{Disc}(Q_2)$ be two distributions such that $\mu_1\, L(R, 0)\, \mu_2$. By definition of 0-lifting, there exist $\mu_1', \mu_1'' \in \mathrm{Disc}(Q_1)$, $\mu_2', \mu_2'' \in \mathrm{Disc}(Q_2)$ such that

$$\begin{aligned}\mu_1 &= (1-0)\mu_1' + 0\mu_1'' \\ \mu_2 &= (1-0)\mu_2' + 0\mu_2'' \\ \mu_1' &\, L(R)\, \mu_2'.\end{aligned}$$

Since $\mu_1 = \mu_1'$ and $\mu_2 = \mu_2'$, then $\mu_1\, L(R)\, \mu_2$.

(⇐) Let $\mu_1 \in \mathrm{Disc}(Q_1), \mu_2 \in \mathrm{Disc}(Q_2)$ be two distributions such that $\mu_1\, L(R)\, \mu_2$. We define $\mu_1' = \mu_1'' = \mu_1$ and $\mu_2' = \mu_2'' = \mu_2$, this implies that

$$\begin{aligned}\mu_1 &= (1-0)\mu_1' + 0\mu_1'', \\ \mu_2 &= (1-0)\mu_2' + 0\mu_2''.\end{aligned}$$

Since $\mu_1' = \mu_1$ and $\mu_2' = \mu_2$, then $\mu_1' L(R) \mu_2'$ and thus $\mu_1\, L(R, 0)\, \mu_2$.

□

The following properties of $L(R, \varepsilon)$ show that this relation is an the equivalence, as the bisimulation needed.

**Proposition 4.6 (reflexivity of ε-lifting).** *For each relation $R$ on $Q$, and each $\varepsilon > 0$, if $R$ is reflexive then $L(R, \varepsilon)$ is reflexive.*

*Proof.* Since $R$ is reflexive, then for each $s \in Q$ we have $s\,R\,s$. We consider a transition of $s$, i.e. $s \xrightarrow{a} \mu$, where $\mu \in \mathrm{Disc}(Q)$ is a measure. If $\varepsilon \geqslant 1$, then $\mu\,L(R,1)\,\mu$ by definition of $\varepsilon$-lifting. If $0 \leq \varepsilon < 1$, then we decompose $\mu$ in $\mu = (1-\varepsilon)\mu + \varepsilon\mu$. The reflexivity of $R$ implies the reflexivity of $L(R)$, i.e. $s\,R\,s$ implies $\mu\,L(R)\,\mu$. Thus $\mu\,L(R,\varepsilon)\,\mu$ for definition of $L(R,\varepsilon)$. ☐

**Proposition 4.7 (symmetry of $\varepsilon$-lifting).** *For each relation from $X$ to $Y$, and each $\varepsilon >= 0$, if $R$ is symmetric then $L(R,\varepsilon)$ is symmetric.*

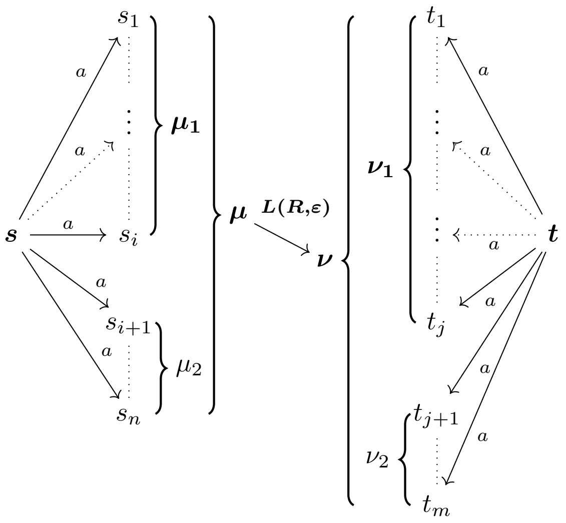
*Proof.* Let $\varepsilon > 0$ and consider two measures $\mu \in \mathrm{Disc}(X)$ and $\nu \in \mathrm{Disc}(Y)$ such that $\mu\,L(R,\varepsilon)\,\nu$. If $\varepsilon \geq 1$, then by definition of $\varepsilon$-lifting, it follows that $\mu\,L(R,\varepsilon)\,\nu$. If $0 \leq \varepsilon < 1$, then we have that there exist $\mu_1, \mu_2, \nu_1, \nu_2$ such that

$$\begin{aligned} \mu =&(1-\varepsilon)\mu_1 + \varepsilon\mu_2, \\ \nu =&(1-\varepsilon)\nu_1 + \varepsilon\nu_2, \\ &\mu_1\,L(R,\varepsilon)\,\nu_1. \end{aligned}$$

Since R is symmetric, by Property 2.1(5) we have that $\mu_1\,L(R,\varepsilon)\,\nu_1$ and thus $\mu\,L(R,\varepsilon)\,\nu$. ☐

### 4.3.2 $\varepsilon$-(bi)simulation

Given a probabilistic system $(Q, A, Tr)$ and two probabilistic states $s$ and $t$, a simulation checks that one state simulates the other. That is if the first state can apply a set $T \subseteq Tr$ of transitions with probability $\mu$, then the second has to simulate the whole set $T$ with a set of transitions with at most an error $\varepsilon$ on its probability of transitions. The $\varepsilon$-lifting on $R$ links the approximation of $\mu$ with the approximation of the second distribution as shown in Diag. 4.2. That is, the action $a$ on the first state $s$ generates a set of resulting states described by the distribution $\mu$, the same happens applying $a$ to the second state $t$ and we obtain the distribution $\nu$. If we apply the $\varepsilon$-lifting to these distributions, we decompose $\mu$ into $\mu_1$ and $\mu_2$ and decompose $\nu$ into $\nu_1$ and $\nu_2$. Since we ignore the error parts of the these decompositions, i.e. $\mu_2$ and $\nu_2$, we relate $\mu_1\,L(R)\,\nu_1$ as shown in Diagram 4.2. Thus we relate also the original distributions by the approximate simulation with error $\varepsilon$ as shown in Diagram 4.3.



(D.4.3)

**Definition 4.8 ($\varepsilon$-simulation).** *Given a probabilistic system $(Q, A, Tr)$ a relation $R$ on $Q$ is an* $\varepsilon$-simulation, *denoted by $s \leqslant_s t$, if for each pair of states $(s,t) \in Q$ such that $s\,R\,t$ and for each $s \xrightarrow{a} \mu \in Tr$, then there exists a transition $t \xrightarrow{a} \nu \in Tr$ such that $\mu\, L(R, \varepsilon)\, \nu$.*

We represent only the $\varepsilon$-simulation and the $R$ relation in Diagram 4.4, where $\approx$ indicates the approximation of the distribution by means of the first part of the decomposition.

$$\begin{tikzcd}
s \arrow[rrr, squiggly, "\boldsymbol{R=\leqslant_s}"] \arrow[d, "a"] \arrow[ddd, bend right, "z"'] & & & t \arrow[d, "a"'] \arrow[ddd, bend left, "z"] \\
\mu^a \arrow[r, phantom, "\approx"] \arrow[d, dotted] & \mu_1^a \arrow[r, squiggly, "L(R)"] \arrow[rr, squiggly, bend right, "L(R,\varepsilon)"'] & \nu_1^a \arrow[r, phantom, "\approx"] & \nu^a \arrow[d, dotted] \\
\cdots \arrow[d, dotted] \arrow[rrr, dashed, squiggly, bend right=10, "L(R,\varepsilon)"] & & & \cdots \arrow[d, dotted] \\
\mu^z \arrow[r, phantom, "\approx"] & \mu_1^z \arrow[r, squiggly, "L(R)"] \arrow[rr, squiggly, bend right, "L(R,\varepsilon)"'] & \nu_1^z \arrow[r, phantom, "\approx"] & \nu^z
\end{tikzcd} \tag{D.4.4}$$

In general we are interesting that $s$ simulates $t$ and vice versa too. $s$ makes a transition with distribution $\mu$, there exists a transition $t \to \nu$ that simulates the first one with error $\varepsilon$. If $s \to \mu$ simulates $t \to \nu$ too, then the relation between these states is an approximate bisimulation. That is, there exists an approximation for $\mu$, that we write $\mu \approx \mu_1$, and an approximation $\nu \approx \nu_1$. These approximations are related by an equivalence relation, that we depict by $\mu_1 \leftrightsquigarrow \nu_1$ in Diagram 4.5.

$$\begin{tikzcd}
s \arrow[r, leftrightsquigarrow, "\boldsymbol{R}"] \arrow[d, "a"] & t \arrow[d, "a"] \\
\mu \approx \mu_1 \arrow[r, leftrightsquigarrow, "\boldsymbol{L(R)}"] & \nu_1 \approx \nu
\end{tikzcd} \tag{D.4.5}$$

But this constraint is too strong for our purposes, thus we relax it imposing only the simulation in one direction and another simulation for the reverse.

**Definition 4.9 ($\varepsilon$-bisimulation).** *Given a relation $R$ if $R$ and $R^{-1}$ are $\varepsilon$-simulations, then $R$ is an* $\varepsilon$-bisimulation.

The difference between the constraint of equivalence relation for simulation and the only existence of a reverse simulation lies in the decompositions of the distributions. In Definition 4.9 there exists a pair of decompositions $\mu \approx \mu_1$ and $\nu \approx \nu_1$, generated by the error $\varepsilon$ in the first simulation, such that $\mu_1$ is related by a lifting relation to $\nu_1$ and we write $\mu_1 \rightsquigarrow \nu_1$. There exists a second pair of decompositions $\mu \approx \mu'$ and $\nu \approx \nu'$, generated by the error $\varepsilon'$ in the second simulation, such that the reverse is verified $\nu' \rightsquigarrow \mu'$, i.e. $\nu'$ is related by a lifting relation to $\mu'$. These two pairs of decompositions generally are different, since the two simulation can allow different errors. For this reason we represent an $\varepsilon$-bisimulation in Diagram 4.6 with two diagrams placed side by side.

$$\begin{tikzcd}
s \arrow[r, rightsquigarrow, "\boldsymbol{R}{=}\leqslant_{\boldsymbol{s}}"] \arrow[d, "a"] \arrow[dd, dotted, bend right] \arrow[ddd, bend right=60, "z"'] & t \arrow[d, "a"'] \arrow[dd, dotted, bend left] \arrow[ddd, bend left=60, "z"] \\
\mu^a \arrow[r, rightsquigarrow, "L(R{,}\varepsilon)"] \arrow[d, dotted, no head] & \nu^a \arrow[d, dotted, no head] \\
\cdots \arrow[r, dashed, rightsquigarrow, "L(R{,}\varepsilon)"] \arrow[d, dotted, no head] & \cdots \arrow[d, dotted, no head] \\
\mu^z \arrow[r, rightsquigarrow, "L(R{,}\varepsilon)"] & \nu^z
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
s \arrow[d, "a"] \arrow[dd, dotted, bend right] \arrow[ddd, bend right=60, "z"'] & t \arrow[l, rightsquigarrow, "\boldsymbol{R^{-1}}{=}\geqslant_{\boldsymbol{s}}"'] \arrow[d, "a"'] \arrow[dd, dotted, bend left] \arrow[ddd, bend left=60, "z"] \\
\mu^a \arrow[d, dotted, no head] & \nu^a \arrow[l, rightsquigarrow, "L(R^{-1}{,}\varepsilon')"'] \arrow[d, dotted, no head] \\
\cdots \arrow[d, dotted, no head] & \cdots \arrow[l, dashed, rightsquigarrow, "L(R^{-1}{,}\varepsilon')"'] \arrow[d, dotted, no head] \\
\mu^z & \nu^z \arrow[l, rightsquigarrow, "L(R^{-1}{,}\varepsilon')"']
\end{tikzcd} \tag{D.4.6}$$

With the notion of approximate bisimulation we define the notion of approximate bisimilarity, which relates two states that are interchangeable.

**Definition 4.10 ($\varepsilon$-bisimilarity).** *Two states $s, t \in Q$ are* $\varepsilon$-bisimilar*, denoted by $s \overset{\varepsilon}{\sim} t$, if there exists a relation $R \subseteq Q \times Q$ such that $R$ is an $\varepsilon$-bisimulation and $s\,R\,t$.*

To emphasise the equivalence relations between states we condense the two diagrams represented in Diag. 4.6 in a single one in Diagram 4.7; in addition we depict the relation $\overset{\varepsilon}{\sim}$ with a dash and dotted line in the above part of the diagram, although this relation be the last one created.

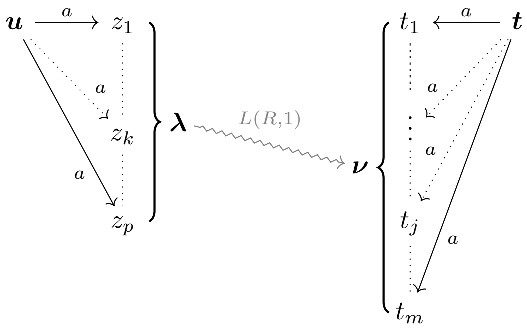
$$\begin{tikzcd}
s_i \arrow[r, leftrightsquigarrow, "\geqslant_{\boldsymbol{s}} = \boldsymbol{R} = \leqslant_{\boldsymbol{s}}"] \arrow[r, dash dot, leftrightarrow, bend left=60, "\overset{\boldsymbol{\varepsilon}}{\sim}"] \arrow[d, "a"] \arrow[dd, dotted, bend right] \arrow[ddd, bend right=60, "z"'] & t_i \arrow[d, "a"'] \arrow[dd, dotted, bend left] \arrow[ddd, bend left=60, "z"] \\
\mu^{i,a} \arrow[r, squiggly, bend left, "L(R,\varepsilon)"] \arrow[d, dotted, no head] & \nu^{i,a} \arrow[l, squiggly, bend left, "L(R^{-1},\varepsilon')"] \arrow[d, dotted, no head] \\
\vdots \arrow[r, dash dot, bend left, "L(R,\varepsilon)"] \arrow[d, dotted, no head] & \vdots \arrow[l, dash dot, bend left, "L(R^{-1},\varepsilon')"] \arrow[d, dotted, no head] \\
\mu^{i,z} \arrow[r, squiggly, bend left, "L(R,\varepsilon)"] & \nu^{i,z} \arrow[l, squiggly, bend left=60, "L(R^{-1},\varepsilon')"]
\end{tikzcd} \tag{D.4.7}$$

*Particular cases*

We have two particular cases related to the errors allowed: $\varepsilon = 0$ and $\varepsilon = 1$. In the first case, given two state $s, t$, the distributions $\mu, \nu$ are already related by $R$-lifting relation, thus there is no needed for decomposing them. The first part of each decomposition, called $mu_1, \nu_1$, is equal to each distribution, i.e. $\mu_1 = \mu$ and $\nu_1 = \nu$, as shown in Diagram 4.8 and proved in Prop. 4.5.

$$\begin{tikzcd}
 & s_1 \arrow[d, dotted, no head] & & & t_1 \arrow[d, dotted, no head] & \\
 & \vdots \arrow[d, dotted, no head] & \Big\}\ \boldsymbol{\mu_1} = \boldsymbol{\mu} \arrow[r, squiggly, "L(R,0)"] \arrow[r, squiggly, bend left=50, "L(R)"] & \boldsymbol{\nu} = \boldsymbol{\nu_1}\ \Big\{ & \vdots \arrow[d, dotted, no head] & \\
\boldsymbol{s} \arrow[ruu, "a"] \arrow[ru, dotted, "a"] \arrow[r, dotted, "a"] \arrow[rd, "a"'] & s_i \arrow[d, dotted, no head] & & & \vdots \arrow[d, dotted, no head] & \boldsymbol{t} \arrow[luu, "a"'] \arrow[lu, dotted, "a"'] \arrow[l, dotted, "a"'] \arrow[ld, "a"] \\
 & s_n & & & t_m &
\end{tikzcd} \tag{D.4.8}$$

In the second case it is not possible to find a decomposition for one of the distributions or to both, however small it is the error allowed. Thus we impose $\varepsilon = 1$, i.e. the maximum error permitted. This means that we generate a relation between any pair of distributions only if the actions of the first states are always simulated by the second state, as shown in Diagram 4.9.


$\boldsymbol{u}$
$a$
$z_1$
$a$
$z_k$
$a$
$z_p$
$\boldsymbol{\lambda}$
$L(R,1)$
$\boldsymbol{\nu}$
$t_1$
$a$
$\boldsymbol{t}$
$a$
$a$
$t_j$
$a$
$t_m$

(D.4.9)

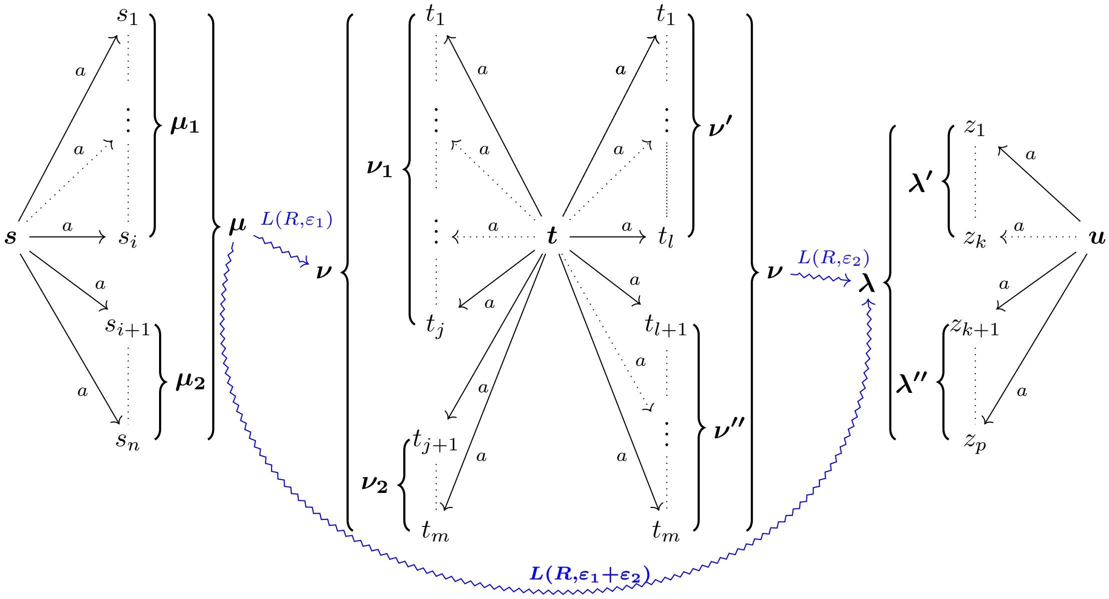
# 5

# From approximated simulations to pseudometrics

In this Chapter we analyse the close relation between the main notions of approximation of the bisimulation for probabilistic systems, i.e. between the approach of (bi)simulation relation and the approach of metrics. In Section 4.3 we recall the notion of approximate (bi)simulation, called $\varepsilon$-bisimulation, that extends the standard notion of (bi)simulation relaxing the definition of simulation introducing an error in each single step. In Section 5.3 we define two pseudometrics, called both $d_{L,R}$, that calculate the smallest error given a pair of states and of distributions. We show that $d_{L,R}$ on distributions is equivalent to the standard Kantorovich metric. In Section 5.5 to study the metrics on probabilistic automata we define a functional transformer and impose some restrictions to be compatible with the models in literature. These transformer is an over-approximation of the metric on probabilistic automata and it is consistent with the literature.

## 5.1 Transitivity of $\varepsilon$-lifting

We have recalled in Chapter 4 that the definition of $L(R, \varepsilon)$, Def. 4.4, verifies the reflex and symmetric properties. The transitive property is not as easy to verify as the reflexivity and the symmetry, since we have inserted the notion of error that propagates at every transition. We consider three states $s, t, u$ and their distributions, respectively, $\mu, \nu$,and $\lambda$, that are represented in Diagram 5.1. The introduction of the error $\varepsilon_1$ allows us to create a lifting on $R$ relation that links $\mu$ with $\nu$; in the same way, error $\varepsilon_2$ allows the creation of lifting on $R$ relation that links $\nu$ with $\lambda$. The state $t$ is a binder that connects the first state with the third by means of the decompositions $\nu_1$ and $\nu'$, which relates, respectively, $\mu$ with $\nu$ and $\nu$ with $\lambda$ by the lifting relation on $R$. In the lifting between $\mu$ and $\lambda$ the error necessary is the sum of the previous two. In the following we give the proof that $L(R, \varepsilon)$ is transitive and, for Prop. 4.6 and Prop. 4.7, it is yet an equivalence.


$s$
$s_1$
$s_i$
$s_{i+1}$
$s_n$
$\boldsymbol{\mu_1}$
$\boldsymbol{\mu_2}$
$\boldsymbol{\mu}$
$L(R,\varepsilon_1)$
$\boldsymbol{\nu}$
$t$
$t_1$
$t_j$
$t_{j+1}$
$t_m$
$t_l$
$t_{l+1}$
$\boldsymbol{\nu_1}$
$\boldsymbol{\nu_2}$
$\boldsymbol{\nu'}$
$\boldsymbol{\nu''}$
$L(R,\varepsilon_2)$
$\boldsymbol{\lambda}$
$\boldsymbol{\lambda'}$
$\boldsymbol{\lambda''}$
$u$
$z_1$
$z_k$
$z_{k+1}$
$z_p$
$a$
$\boldsymbol{L(R,\varepsilon_1+\varepsilon_2)}$

(D.5.1)

**Proposition 5.1 (Transitivity of $\varepsilon$-lifting).** *Given a probabilistic system $(Q, A, Tr)$, let $R$ be a transitive relation on $Q$ and $\mu, \nu, \lambda \in Disc(Q)$ are probability measures. For all $\varepsilon_1, \varepsilon_2 \in [0,1]$ if $\mu\, L(R, \varepsilon_1)\, \nu$ and $\nu\, L(R, \varepsilon_2)\, \lambda$ then $\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \lambda$.*

We give a simplify diagram to represent the transitive property, in Diagram 5.2 we abstract from the internal states.

$$\begin{tikzcd}
s \arrow[rrr, rightsquigarrow, "R"] \arrow[rrrrrr, rightsquigarrow, bend left, "R"] \arrow[d, "a"] & & & t \arrow[rrr, rightsquigarrow, "R"] \arrow[d, "a"'] & & & u \arrow[d, "a"'] \\
\mu \arrow[rrr, rightsquigarrow, bend right, "\boldsymbol{L(R,\varepsilon_1)}"'] \arrow[rrrrrr, rightsquigarrow, bend right=40, "\boldsymbol{L(R,\varepsilon_1+\varepsilon_2)}"'] & \approx \mu_1 \arrow[r, rightsquigarrow, "L(R)"] & \nu_1 \approx & \nu \arrow[rrr, rightsquigarrow, bend right, "\boldsymbol{L(R,\varepsilon_2)}"'] & \approx \nu' \arrow[r, rightsquigarrow, "L(R)"] & \lambda' \approx & \lambda
\end{tikzcd}$$

(D.5.2)

*Proof.* We consider several cases that depends on the values of the errors.

$\varepsilon_1 + \varepsilon_2 \geqslant 1$: we have $\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \lambda$ for Definition 4.4.

$\varepsilon_1 = \varepsilon_2 = 0$: by hypothesis we have $\mu\, L(R, 0)\, \nu$ and $\nu\, L(R, 0)\, \lambda$, that are equal to $\mu\, L(R)\, \nu$ and $\nu\, L(R)\, \lambda$ for Proposition 4.5. The lifting of $R$ is transitive (Prop. 3.28), thus we infer $\mu\, L(R)\, \lambda$ and with Prop. 4.5 we obtain $\mu\, L(R, 0)\, \lambda$.

$\varepsilon_1 = 0, \varepsilon_2 \in (0, 1)$: by hypothesis we have $\mu\, L(R, 0)\, \nu$ and $\nu\, L(R, \varepsilon_2)\, \lambda$. From first one with Prop. 4.5 we obtain $\mu\, L(R)\, \nu$. From the second one with definition of lifting we have

$$\begin{aligned} \nu =& (1-\varepsilon_2)\nu_1 + \varepsilon_2\nu_2, \\ \lambda =& (1-\varepsilon_2)\lambda_1 + \varepsilon_2\lambda_2, \\ & \nu_1\, L(R)\, \lambda_1. \end{aligned}$$

We apply Prop. 3.5(7) of [62] to $\lambda = (1-\varepsilon_2)\lambda_1 + \varepsilon_2\lambda_2$ and $\nu_1\, L(R)\, \lambda_1$ and obtain $\mu = (1-\varepsilon)\mu_1 + \varepsilon\mu_2$ such that $\mu_1\, L(R)\, \lambda_1$ and $\mu_2\, L(R)\, \lambda_2$. We have a decomposition for $\mu$ with $\varepsilon_1$, a decomposition for $\lambda$ with $\varepsilon_2$ and $\mu_1\, L(R)\, \lambda_1$. This is the definition of approximation, thus $\mu\, L(R\varepsilon_2)\, \lambda$, that is $\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \lambda$ since $\varepsilon_1 = 0$.

$\varepsilon_1 \in (0, 1), \varepsilon_2 = 0$: similar to the previous.

$\varepsilon_1 + \varepsilon_2 < 1$: with $\varepsilon_1 \neq 0$ and $\varepsilon_2 \neq 0$. By hypothesis we have $\mu\, L(R, \varepsilon_1)\, \nu$ and $\nu\, L(R, \varepsilon_2)\, \lambda$, thus we have a decomposition for $\mu$ and $\nu$ with error $\varepsilon_1$ and a decomposition for $\nu$ and $\lambda$ with error $\varepsilon_2$. We highlight the two decomposition for $\nu$:

$$\begin{aligned} \nu =& (1-\varepsilon_1)\nu_1 + \varepsilon_1\nu_2, \\ \nu =& (1-\varepsilon_2)\nu' + \varepsilon_2\nu''. \end{aligned}$$

We want to find a decomposition for $\nu$ such that it includes both the errors, thus we consider $\varepsilon = \max\{\varepsilon_1, \varepsilon_2\}$ and

$$\begin{aligned} \nu_1' =& \frac{\min\{(1-\varepsilon_1)\nu_1, (1-\varepsilon_2)\nu'\}}{1-\varepsilon} \\ \nu_2'' =& \frac{\max\{\varepsilon_1\nu_2, \varepsilon_2\nu''\}}{\varepsilon}. \end{aligned}$$

$\nu_1'$ is a measures, since $\nu_1, \nu'$ are measures, $(1-\varepsilon) = 0$ only if $\varepsilon = 1$, and

$$\nu_1'(Q) = \frac{\min\{(1-\varepsilon_1)1, (1-\varepsilon_2)1\}}{1-\varepsilon} = \frac{1-\max\{\varepsilon_1, \varepsilon_2\}}{1-\varepsilon} = \frac{1-\varepsilon}{1-\varepsilon} = 1.$$

Similar $\nu_2''$ is a measure.

By definition of $\nu_1'$ and $\nu_2''$ we can easily infer that $\nu = (1-\bar{\varepsilon})\nu_1' + \bar{\varepsilon}\nu_2''$, since for all $q \in Q$ we have

$$\min\{(1-\varepsilon_1)\nu_1(q), (1-\varepsilon_2)\nu'(q)\} = (1-\varepsilon_1)\nu_1(q)$$

if and only if

$$\max\{\varepsilon_1\nu_2(q), \varepsilon_2\nu''(q)\} = \varepsilon_1\nu_2(q).$$

We consider $\bar{\varepsilon} = \varepsilon_1 + \varepsilon_2$. The inequalities $\nu = (1-\bar{\varepsilon})\nu_1' + \bar{\varepsilon}\nu_2''$ and $\bar{\varepsilon} > \varepsilon$ satisfy the conditions of Prop. 3.5(3) of [62], thus we have

$$\nu = (1-\bar{\varepsilon})\nu_1' + \bar{\varepsilon}\nu_3, \text{ where } \nu_3 = \left(1 - \frac{\varepsilon_1}{\bar{\varepsilon}}\right)\nu_1' + \frac{\varepsilon_1}{\bar{\varepsilon}}\nu_2''.$$

Since $\bar{\varepsilon} > \varepsilon$ and, by hypothesis $\mu\, L(R, \varepsilon_1)\, \nu$ and $\nu\, L(R, \varepsilon_2)\, \lambda$, then we can infer $\mu_1\, L(R)\, \nu_1'$ and $\nu_1'\, L(R)\, \lambda'$. For Prop. 3.28, i.e. transitivity of $L(R)$, we have $\mu_1\, L(R)\, \lambda'$ and thus $\mu\, L(R, \bar{\varepsilon})\, \lambda$, which is $\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \lambda$.

□

## 5.2 $\varepsilon$-relation

In this section we generalise the notion of equivalence between states of a probabilistic system. We relax the constraint included in bisimulation definition, Def. 4.9, it requires that the $\varepsilon$-lifting relation between distributions is verified for every pair of states of the system.

**Definition 5.2 ($\varepsilon$-relation).** *Given a relation $R \subseteq Q \times Q$ two states $s, t \in Q$ are in relation $s \overset{\varepsilon}{\sim}_R t$ if*

- *for each $s \xrightarrow{a} \mu \in Tr$ there exists $t \xrightarrow{a} \nu \in Tr$ such that $\mu\, L(R, \varepsilon)\, \nu$,*
- *for each $t \xrightarrow{a} \nu \in Tr$ there exists $s \xrightarrow{a} \mu \in Tr$ such that $\nu\, L(R, \varepsilon')\, \mu$.*

We represent the $\varepsilon$-relation with Diagram 5.3. Several arrows start from state $s$, each one has a label that denotes the transition represented. These arrows represent all the possible transitions from $s$, specularly these transitions are possible also from $t$. Each transition generates a distribution from $s$ which is in $\varepsilon$-lifting relation with the distribution generated from $t$. This constraint is verified for each transition, thus the states $s$ and $t$ are in $\varepsilon$-relation.

$$\begin{tikzcd}
s \arrow[rr, squiggly, "R"] \arrow[rr, dash dot, bend left, leftrightarrow, "\overset{\varepsilon}{\sim}_{\boldsymbol{R}}"] \arrow[d, "a"] \arrow[dd, dotted, bend right] \arrow[ddd, bend right=60, "z"'] & & t \arrow[d, "a"'] \arrow[dd, dotted, bend left] \arrow[ddd, bend left=60, "z"] \\
\mu^a \arrow[rr, squiggly, bend left, "L(R,\varepsilon)"] \arrow[d, dotted, no head] & & \nu^a \arrow[ll, squiggly, bend left, "L(R^{-1},\varepsilon')"] \arrow[d, dotted, no head] \\
\cdots \arrow[rr, dash dot, bend left, "L(R,\varepsilon)"] \arrow[d, dotted, no head] & & \cdots \arrow[ll, dash dot, bend left, "L(R^{-1},\varepsilon')"] \arrow[d, dotted, no head] \\
\mu^z \arrow[rr, squiggly, bend left, "L(R,\varepsilon)"] & & \nu^z \arrow[ll, squiggly, bend left, "L(R^{-1},\varepsilon')"]
\end{tikzcd} \tag{D.5.3}$$

Nevertheless, $\overset{\varepsilon}{\sim}_R$ is not an $\varepsilon$-bisimilarity: the relation is verified only for the single pair of states $(s,t)$. The above diagram and Diag. 4.7 differ for the symbols in the above parts; but they represent two relations very different. This difference is visible by the subscripts in Diag. 4.7, where $s_i$ represents an element of the set of states of the system, as well as $t_i$, that are not present in Diag. 5.3.

In the above definition there are two particular cases depending on the values of the error $\varepsilon$, thus we introduce some notations to easily identify them.

$\varepsilon = 0$: if there is a direct simulation between two states and there is no need to insert an error, then we write $\overset{0}{\sim}_R$.

$\varepsilon = 1$: if the actions of a set of transitions of a state is equal to the actions of the set of transitions of the simulating state, but the distributions are never in relation either by introducing any error. Then we write $\overset{1}{\sim}_R$.

Problem: the $\varepsilon$-relation defined seems to be not transitive, since we have inserted an error that propagates on transitions. We show the property of light transitivity for $\varepsilon$-relation by using Def. 5.1.

**Proposition 5.3.** *Given $(Q, A, T)$ a probabilistic system, let $R$ be a transitive relation on $Q$ and $p, s, q \in Q$ are states of the system. For all $\varepsilon_1, \varepsilon_2 \in [0,1]$ if $p \overset{\varepsilon_1}{\sim}_R s$ and $s \overset{\varepsilon_2}{\sim}_R q$, then we have $p \overset{\varepsilon_3}{\sim}_R q$ where $\varepsilon_3 = \varepsilon_1 + \varepsilon_2$.*

Before proving the proposition we give a representation of the transitive property with Diagram 5.4. To avoid the representation of too much information, we have eliminated the approximations of the distributions with respect to Diag. 5.2. Here different errors, as $\varepsilon_1, \varepsilon_2$, represent different decompositions of also a single distribution.

$$\begin{tikzcd}
\color{blue}{p} \arrow[rr, leftrightarrow, squiggly, "R"] \arrow[rr, leftrightarrow, dashed, bend left, "\overset{\boldsymbol{\varepsilon_1}}{\sim}_{\boldsymbol{R}}"] \arrow[rrrr, leftrightarrow, dashed, bend left=50, "\overset{\boldsymbol{\varepsilon_1+\varepsilon_2}}{\sim}_{\boldsymbol{R}}"] \arrow[d, "a"] \arrow[dd, bend right, "a"'] & & \color{blue}{s} \arrow[rr, leftrightarrow, squiggly, "R"] \arrow[rr, leftrightarrow, dashed, bend left, "\overset{\boldsymbol{\varepsilon_2}}{\sim}_{\boldsymbol{R}}"] \arrow[d, "a"] \arrow[dd, bend left, "a"] & & \color{blue}{q} \arrow[d, "a"] \arrow[dd, bend left, "a"] \\
\mu \arrow[rr, squiggly, "{L(R,\varepsilon_1)}"] \arrow[rrrr, squiggly, bend right, "{L(R,\varepsilon_1+\varepsilon_2)}"'] & & \nu \arrow[rr, squiggly, "{L(R,\varepsilon_2)}"] & & \xi \\
\mu & & \nu \arrow[ll, squiggly, "{L(R^{-1},\varepsilon_1')}"'] & & \xi \arrow[ll, squiggly, "{L(R^{-1},\varepsilon_2')}"'] \arrow[llll, squiggly, bend left, "{L(R^{-1},\varepsilon_1'+\varepsilon_2')}"]
\end{tikzcd} \tag{D.5.4}$$

*Proof.* By hypothesis we have

1. $p \overset{\varepsilon_1}{\sim}_R s$, thus for each $p \xrightarrow{a} \mu$ there exists a state $s \in Q$ such that $s \xrightarrow{a} \nu$ and $\mu\, L(R, \varepsilon_1)\, \nu$.
2. $s \overset{\varepsilon_2}{\sim}_R q$, thus for each $s \xrightarrow{a} \nu$ there exists a state $q \in Q$ such that $q \xrightarrow{a} \xi$ and $\nu\, L(R, \varepsilon_2)\, \xi$.

By Proposition 5.1, that is the transitivity of $\varepsilon$-lifting, we have $\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \xi$ for each $p \xrightarrow{a} \mu$ with the existence of $q \xrightarrow{a} \xi$. Since $\overset{\varepsilon_1}{\sim}_R$ is symmetric, we have also $\nu\, L(R, \varepsilon_1)\, \mu$ and $\xi\, L(R, \varepsilon_2)\, \nu$. Thus the sequence $\xi\, L(R, \varepsilon_2)\, \nu \circ \nu\, L(R, \varepsilon_1)\, \mu$ gives us $\xi\, L(R, \varepsilon_1 + \varepsilon_2)\, \mu$ for each $q \xrightarrow{a} \xi$ with the existence of $p \xrightarrow{a} \mu$.

$\mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \xi$ and $\xi\, L(R, \varepsilon_1 + \varepsilon_2)\, \mu$ leads to the definition of $p \overset{\varepsilon_1 + \varepsilon_2}{\sim}_R q$. □

## 5.3 Probabilistic metrics with ε-lifting

In this section we define a metric on probabilistic automata, rather a pseudometric, since we do not need that the difference between two distinct elements has a value greater of zero. A pseudometric differs from an ordinary metric since different elements can have distance 0. The pseudodistance between states is a real number between 0 and 1, it is used to express the similarity of the behaviour of those states. Using a terminology introduced by Sangiorgi [12], we say that a relation between processes $P_1$ progresses to $P_2$ if for every pair of processes in $P_1$, every transition from one process is matched by a transition from the other, and the derivative processes are related by $P_2$. Thus a metric $d_1$ on states progresses to a metric $d_2$ on distributions over states if, for all processes at $d_1$-distance $\varepsilon$, every transition from one process is matched by a transition from the other and the resulting distributions are at $d_2$-distance at most $\varepsilon$. Then $d_1$ is a bisimulation metric if it progresses to its own lifting $L(d_1)$ on distributions. Among the bisimulation metrics, those based on the Kantorovich lifting are the most popular.

**Definition 5.4 (Pseudometric).** *A function $d\colon \Omega \times \Omega \to \mathbb{R}$ is a* pseudometric *if*

1. *(nonnegativity) for all $x, y \in \Omega$ we have $d(x, y) \geqslant 0$*
2. *(reflexivity) for all $x \in \Omega$ we have $d(x, x) = 0$*
3. *(symmetry) for all $x, y \in \Omega$ we have $d(x, y) = d(y, x)$*
4. *(triangle inequality) for all $x, y, z \in \Omega$ we have $d(x, y) + d(y, z) \geqslant d(x, z)$*

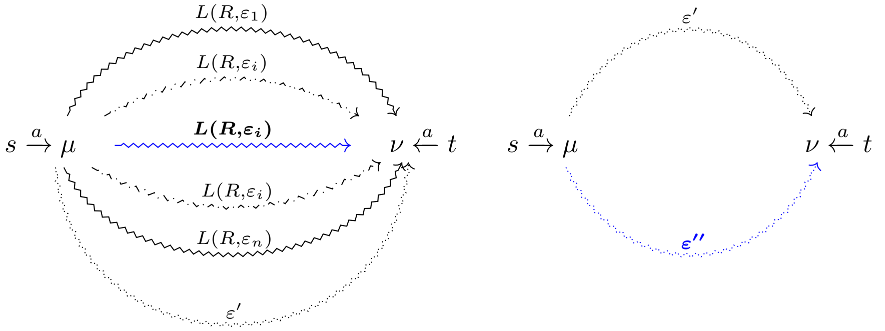
We deduce $d(x, y) \geqslant 0$ from item 1. and 3. Furthermore, if $d$ also satisfies $d(x, y) > 0$ when $x \neq y$, then $d$ is a metric.

We introduce two pseudometrics based on the relation of $\varepsilon$-lifting with the purpose of studying the approximate probabilistic bisimilarity. The first pseudometric that we create is a distance between discrete distributions, we calculate the minimum error that verifies $\varepsilon$-simulation $L(R, \varepsilon)$.

**Definition 5.5 (Distance $d_{L,R}$).** *Let $(Q, A, Tr)$ be a probabilistic system and $R$ an equivalence relation on $Q$, for each $(\mu, \nu) \in Disc(Q)$ the function 1-bounded $d_{L,R}(\mu, \nu)\colon Disc(Q) \times Disc(Q) \to \mathbb{R}$ is defined by*

$$d_{L,R}(\mu, \nu) = \inf_{\varepsilon \in [0,1]} \mu\ L(R, \varepsilon)\ \nu.$$

In Diagram 5.5 we give a representation of $d_{L,R}(\mu, \nu)$. At the left hand side we have represented with an arrow each approximate lifting with indicated the error, for example $\varepsilon_1$. The blue arrow represents the lifting with the minimum error, which is the distance $d_{L,R}$. The dotted line, with the label $\varepsilon'$, represents an error which is a lower bound, for accuracy is the greatest lower bound. This element is the infimum and is the value of $d_{L,R}$, if it is not possible to create an $\varepsilon$-lifting relation between the two distributions given. This case is shown in the right hand side of Diagram 5.5.



(D.5.5)

**Proposition 5.6.** *Given a probabilistic system $(Q, A, Tr)$ if $R \subseteq Q \times Q$ is an equivalence relation, then $d_{L,R}(\mu, \nu)$ is a pseudometric.*

*Proof.* 1. (nonnegativity) is the definition of $\varepsilon$-simulation (Def. 4.4)

2. (reflexivity) $R$ is an equivalence relation, then $L(R, \varepsilon)$ is reflexive for Proposition 4.6. For each $s \in Q$ such that $s \xrightarrow{a} \mu$ we have $\mu\, L(R, \varepsilon)\, \mu$, in particular $\varepsilon = 0$ verifies $\mu\, L(R, 0)\, \mu$. Since the distance is non negative, the minimum is $\varepsilon = 0$ and thus $d_{L,R}(\mu, \mu) = 0$.
3. (symmetry) Since $R$ is an equivalence relation, then for Proposition 4.7 $L(R, \varepsilon)$ is symmetric. We generate $S = \{\, \varepsilon \in [0, 1] \mid \mu\, L(R, \varepsilon)\, \nu \text{ and } \nu\, L(R, \varepsilon)\, \mu \,\}$. Since both $d_{L,R}(\mu, \nu)$ and $d_{L,R}(\nu, \mu)$ calculate the infimum on $S$, the result must be equal and thus $d_{L,R}(\mu, \nu) = d_{L,R}(\nu, \mu)$.
4. (triangle inequality) We consider $\mu\, L(R, \varepsilon_1)\, \nu$ and $\nu\, L(R, \varepsilon_2)\, \eta$, by Prop. 5.1 we have $\mu\, L(R, \varepsilon_1{+}\varepsilon_2)\, \nu$. Let $S_1 = \{\, \varepsilon_1 \mid \mu\, L(R, \varepsilon_1)\, \nu \,\}$, $S_2 = \{\, \varepsilon_2 \mid \nu\, L(R, \varepsilon_2)\, \eta \,\}$ and $S = \{\, \varepsilon_1 + \varepsilon_2 \mid \mu\, L(R, \varepsilon_1 + \varepsilon_2)\, \nu \,\} = S_1 \cup S_2$ are sets. We consider $\inf S_3 = \inf\{S_1 \cup S_2\} = \inf S_1 + \inf S_2$. For definition of metric we

have $d_{L,R}(\mu,\nu) + d_{L,R}(R\,\nu,\eta) = d_{L,R}(\mu,\eta)$, thus the triangle inequality $d_{L,R}(\mu,\nu) + d_{L,R}(R\,\nu,\eta) \geqslant d_{L,R}(\mu,\eta)$ is verified.

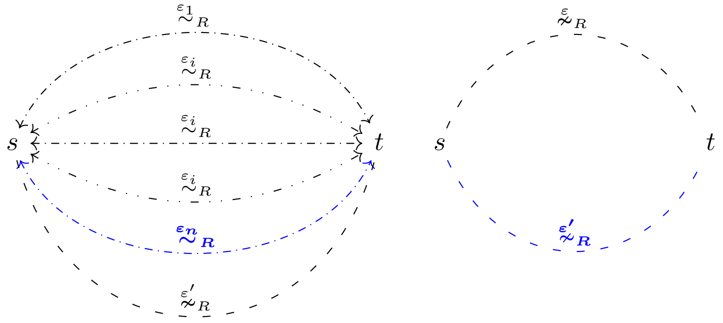
□

The second pseudometric that we create is a distance between states. We calculate the minimum error that verifies $\varepsilon$-simulation $L(R,\varepsilon)$.

**Definition 5.7 (Distance $d_{L,R}$).** *Given a probabilistic system $(Q,A,Tr)$ and a relation $R \subseteq Q \times Q$ a function $d_{L,R}\colon Q \times Q \to \mathbb{R}$ is defined for all $s,t \in Q$ as*

$$d_{L,R}(s,t) = \begin{cases} 1 & \text{if } s \xrightarrow{a} \mu \text{ and } t \overset{a}{\nrightarrow} \\ \inf_{\varepsilon\in[0,1]} s \overset{\varepsilon}{\sim}_R t & \text{otherwise.} \end{cases}$$

In this definition on states we need to consider if all the actions of the first state are simulated by the second state; for each $a$ such that $s \xrightarrow{a} \mu$ we check if there exists $t \xrightarrow{a} \nu$ that simulate the transition of $s$ with action $a$, otherwise $t \overset{a}{\nrightarrow}$ and we impose value 1 to the distance. This process is not necessary for Def. 5.5, since $\nu$ is a parameter and thus every action is simulated. In Diagram 5.6 we represent Def. 5.7, where the arrows with labels $\nsim$ represent the lower or upper bound. Every arrow represents a set of $L(R,\varepsilon)$ and $L(R^{-1},\varepsilon'')$ relations, i.e. represents a whole diagram as Diag. 5.3.


$\overset{\varepsilon_1}{\sim}_R$
$\overset{\varepsilon_i}{\sim}_R$
$\overset{\varepsilon_i}{\sim}_R$
$\overset{\varepsilon_i}{\sim}_R$
$\overset{\varepsilon_n}{\sim}_R$
$\overset{\varepsilon'}{\nsim}_R$
$s$
$t$
$\overset{\varepsilon}{\nsim}_R$
$s$
$t$
$\overset{\varepsilon'}{\nsim}_R$

(D.5.6)

**Theorem 5.8.** *Given a probabilistic system $(Q,A,Tr)$, if $R \subseteq Q \times Q$ is an equivalence relation then $d_{L,R}$ is a pseudometric.*

*Proof.* First we consider a particular case, when two states $s,t \in Q$ execute the same action with very different distributions, then $\varepsilon = 1$ and the approximate bisimilarity is verified. For the remaining cases we verify the axioms of the definition of pseudometric, Definition 5.4.

1. (nonnegativity) is the definition of $\varepsilon$-bisimulation (Def. 4.4)
2. (reflexivity) since $R$ is a bisimulation, that defines the two-bisimilar equivalence $\overset{\varepsilon}{\sim}_R$, and it is reflexive, then for all $x \in Q$ we have $x \overset{0}{\sim}_R x$ and $x \overset{\varepsilon}{\nsim}_R x$.

3. (symmetry) for all pair $x, y \in Q$ we show that $d_{L,R}(x,y) = d_{L,R}(y,x)$. Let are $S_1 = \{\, \varepsilon \mid x \overset{\varepsilon}{\sim}_R y \,\}$ and $S_2 = \{\, \varepsilon \mid y \overset{\varepsilon}{\sim}_R x \,\}$. We suppose there exists $\varepsilon' \in [0,1]$ such that $\varepsilon' \in S_2$ and $\varepsilon' \notin S_1$, thus $y \overset{\varepsilon'}{\sim}_R x$. For definition the relation $R$ is a $\varepsilon'$-bisimulation, i.e. $R$ and $R^{-1}$ are two $\varepsilon'$-simulation. Since $R^{-1}$ is an $\varepsilon'$-simulation, then $y \overset{\varepsilon'}{\sim}_{R^{-1}} x$ that is equivalent to $x \overset{\varepsilon'}{\sim}_R y$. This is in contrast with the hypothesis $\varepsilon' \notin S_1$, where $x \overset{\varepsilon'}{\nsim}_R y$. There is no element that belongs to $S_2$ and does not belong to $S_1$, thus $S_2 = S_1$ and $\inf S_2 = \inf S_1$, i.e. $d_{L,R}(x,y) = d_{L,R}(y,x)$.
4. (triangle inequality) Let $S_{xy} = \{\, \varepsilon_1 \mid x \overset{\varepsilon_1}{\sim}_R y \,\}$ and $S_{yz} = \{\, \varepsilon_2 \mid y \overset{\varepsilon_2}{\sim}_R z \,\}$ are sets. We consider $x \overset{\varepsilon_1}{\sim}_R y$ and $y \overset{\varepsilon_2}{\sim}_R z$, by Prop. 5.3 we have $x \overset{(\varepsilon_1+\varepsilon_2)}{\sim}_R z$. Thus we construct the set $S_{xz} = S_{xy} \cup S_{yz} = \{\, \varepsilon_1 + \varepsilon_2 \mid x \overset{\varepsilon_1}{\sim}_R y \text{ and } y \overset{\varepsilon_2}{\sim}_R z \,\} = \{\, \varepsilon_1 + \varepsilon_2 \mid x \overset{\varepsilon_1+\varepsilon_2}{\sim}_R z \,\}$. We consider the elements $e_1 = \inf S_{xy}$, $e_2 = \inf S_{yz}$, and $e_1 + e_2 = \inf S_{xz} = \inf\{S_{xy} \cup S_{yz}\} = \inf S_{xy} + \inf S_{yz}$. For Definition 5.7 we have $d_{L,R}(x,y) + d_{L,R}(y,z) = d_{L,R}(x,z)$, the triangle inequality $d_{L,R}(x,y) + d_{L,R}(y,z) \geqslant d_{L,R}(x,z)$ is verified.

□

## 5.4 Equivalence of pseudometrics

The pseudometric with approximation on measures is equivalent with the widely used Monge-Kantorovich metric, that we have described in Section 3.4.

**Theorem 5.9.** *Given a probabilistic system $(Q, A, Tr)$, for each relation $R$ and for each probabilistic distributions $\mu, \nu \in Disc(Q)$ we have*

$$d_{L,R}(\mu,\nu) = d_K(\mu,\nu). \tag{5.7}$$

*Proof.* Since we are handing with discrete distributions, we consider Formula 3.14 to calculate the Kantorovich metric. We split the proof in two step

1. $d_{L,R}(\mu,\nu) \geqslant d_K(\mu,\nu)$
2. $d_{L,R}(\mu,\nu) \leqslant d_K(\mu,\nu)$.

1. We consider $\mu\, L(R,\varepsilon)\, \nu$, by definition of $L(R,\varepsilon)$ $\mu, \nu$ can be decomposed in $\mu = (1-\varepsilon)\mu_1 + \varepsilon\mu_2$ and $\nu = (1-\varepsilon)\nu_1 + \varepsilon\nu_2$ such that $\mu_1\, L(R)\, \nu_1$. By $L(R)$ there exists a weighting function $w\colon Q \times Q \to [0,1]$ such that $\sum_{s\in Q} w(s,t) = \nu_1(t)$, $\sum_{t\in Q} w(s,t) = \mu_1(s)$ and, for each pair $(s,t) \in Q$, $w(s,t) > 0$ implies $s\, R\, t$. We construct an element $\sum_{x\in Q}\sum_{y\in Q} d(x,y) \cdot m(x,y)$, where we choose $d = d_R$ (Def.3.13) that defines the metric space $(\Omega, d_R)$. We defining $m$ by

$$m = (1-\varepsilon)m_1(s,t) + \varepsilon m_2(s,t),$$

where $m_1(s,t) = w(s,t)$ and $m_2(s,t) = \mu_2(s) \cdot \nu_2(t)$. With this definition $m$ is an upper bound of $w$ compatibles with $(\mu, \nu)$. The functions $m_1, m_2$ are Borel probability measures and verify the three conditions of Def. 3.9. Since we are in the discrete case, then we simplify the proof as following. We have $\sum_{s\in Q} m_1(s,t) = \sum_{s\in Q} w(s,t) = \nu_1(t)$ and $\sum_{t\in Q} m_1(s,t) = \sum_{t\in Q} w(s,t) = \mu_1(s)$, then $m_1 \in M_1(\mu_1, \nu_1)$. $\sum_{s\in Q} m_2(s,t) = \nu_2(t)$, $\sum_{t\in Q} m_2(s,t) = \mu_2(s)$, then $m_2 \in M_2(\mu_2, \nu_2)$. Since $\mu$ and $\nu$ are compositions of, respectively, $\mu_1, \mu_2$ and $\mu_1, \mu_2$, then $m = (1-\varepsilon)m_1(s,t) + \varepsilon m_2(s,t)$ is a Borel measures, i.e. $m \in M(\mu, \nu)$. Now we calculate $\sum_{x\in Q} \sum_{y\in Q} d_R(x,y) \cdot m(x,y)$.

$$\begin{aligned}
\sum_{s,t\in Q} d_R(s,t)m(s,t) &= \sum_{s,t\in Q} d_R(s,t)\left((1-\varepsilon)m_1(s,t) + \varepsilon m_2(s,t)\right) \\
&= (1-\varepsilon)\sum_{s,t\in Q} d_R(s,t)m_1(s,t) + \varepsilon \sum_{s,t\in Q} d_R(s,t)m_2(s,t) \\
&= 0 + \varepsilon \sum_{s,t\in Q} d_R(s,t)m_2(s,t) \\
&= 0 + \varepsilon \sum_{s,t\in Q} 1\ \mu_2(s) \cdot \nu_2(t) \\
&= 0 + \varepsilon \cdot 1
\end{aligned}$$

The first sum is always 0 since when $d_R(s,t) = 0$ then $m_1(s,t) = 1$ and when $d_R(s,t) = 1$ then $m_1(s,t) = 0$.
For each $\varepsilon$ such that $\mu\, L(R,\varepsilon)\, \nu$, we construct $\sum_{s,t\in Q} d_R(s,t)m(s,t) = \varepsilon$. We have considered an $\varepsilon$ which is a lower bound of $d_{L,R}$, we have constructed a corresponding element in the set of $d_K$ with value $\leqslant \varepsilon$. If $d_{L,R} = \varepsilon$, then we $d_K(\mu,\nu) \leqslant \varepsilon$. Since $d_K(\mu,\nu) \leqslant \varepsilon = d_{L,R}(\mu,\nu)$, we conclude that $d_K(\mu,\nu) \leqslant d_{L,R}(\mu,\nu)$.

2. We consider $m \in M(\mu,\nu)$ that defines $\sum_{s,t\in Q} d_R(s,t)m(s,t) = \varepsilon$. We recall that the distance $d_R(s,t)$ is defined 0 if $s\,R\,t$ and 1 otherwise. Thus

$$\begin{aligned}
\sum_{s,t\in Q} d_R(s,t)m(s,t) &= \varepsilon \\
&= \sum_{s,t.s\,R\,t} d_R(s,t)m_1(s,t) + \sum_{s,t.(s,t)\notin R} d(s,t)m_2(s,t) \\
&= 0 + \sum_{s,t.(s,t)\notin R} 1 m_2(s,t).
\end{aligned}$$

We define $\mu, \nu$ using the affinity/compatibility between the conditions on the marginals $\mu, \nu$ of $M$ and the weighting function $w$ defined by $L(R,\varepsilon)$. We define a first set $M_1(\mu_1, \nu_1)$ with elements $m_1$ such that

$$m_1(s,t) = \begin{cases} 1 - m(s,t) = 1 - \varepsilon & \text{if } s\,R\,t \\ 0 & \text{otherwise.} \end{cases}$$

We define a second set $M_2(\mu_2, \nu_2)$ with elements $m_2$ as

$$m_2(s,t) = \begin{cases} 0 & \text{if } (s,t) \notin R \\ m_2(s,t) = m(s,t) = \varepsilon & \text{otherwise.} \end{cases}$$

We use the projections, respectively, on the first state $\pi_s$ and on the second state $\pi_t$ to define

$$\mu_1 = \frac{\pi_s(m_1)}{(1-\varepsilon)}, \quad \nu_1 = \frac{\pi_t(m_1)}{(1-\varepsilon)}, \qquad \mu_2 = \frac{\pi_s(m_2)}{\varepsilon}, \quad \nu_2 = \frac{\pi_t(m_2)}{\varepsilon}.$$

We calculate $\mu$ and $\nu$ as the compositions

$$\begin{aligned} (1-\varepsilon)\mu_1 + \varepsilon\mu_2 &= \pi_s(m_1) + \pi_s(m_2) \\ &= \pi_s(m) \\ &= \mu \end{aligned}$$

and

$$\begin{aligned} (1-\varepsilon)\nu_1 + \varepsilon\nu_2 &= \pi_t(m_1) + \pi_t(m_2) \\ &= \pi_t(m) \\ &= \nu. \end{aligned}$$

Since $m_1(s,t) + m_2(s,t) = (1-\varepsilon) + \varepsilon = 1$, then $\mu, \nu \leqslant 1$ and thus they are distributions. We have define $\mu, \nu$ such that $\mu\, L(R, \varepsilon)\, \nu$ is verified.
We consider the particular case where $\varepsilon$ is the greatest lower bound of $d_K$, i.e. $d_K(\mu,\nu) = \varepsilon$. The corresponding element in $d_{L,R}$ has value $\varepsilon$ and belongs to $\mu\, L(R,\varepsilon)\, \nu$, thus $d_{L,R}(\mu,\nu) = \varepsilon$. It follows that $d_{L,R}(\mu,\nu) = \varepsilon \leqslant d_K(\mu,\nu)$ and $d_{L,R}(\mu,\nu) \leqslant d_K(\mu,\nu)$.

□

## 5.5 Bisimilarity as fixed-point of the operator F

In this section we recast the iterator operator of the fixed point definition of bisimulation recalled in Section 3.3.3. In particular we consider the De Alfaro et al. operator $H^{\text{MDP}}_{\text{post}}$ (Def. 3.52). De Alfaro et al.'s operator works with Markov Decision Process (MDP), this implies that the sets of actions of two states are always equivalent. The operator we want generate is based on probabilistic automata, here the set of actions of a state could not be simulated by the set of actions of another state of the automata. For this reason we create a pseudo-evaluation $v_a(s)$, that checks if the label of an action is enabled in a state, and an equivalence of actions $s \equiv_a t$, that impose the value 1 to the states with different sets of actions.

**Definition 5.10 (Action evaluation).** *Given a probabilistic system* $(Q, A, Tr)$ *and an action* $a \in A$*, we define the function* action evaluation $v_a \colon Q \to (0, 1)$ *by*

$$v_a(s) = \begin{cases} 1 & \textit{if } s \xrightarrow{a} \mu \\ 0 & \textit{if } s \overset{a}{\not\to} \end{cases}$$

**Definition 5.11 (Equivalence of actions).** *Given a probabilistic system* $(Q, A, Tr)$ *and* $s, t \in Q$ *we define the* equivalence of actions*, for each action* $a \in A$*, by*

$$s \equiv_a t = |v_a(s) - v_a(t)|\,.$$

*We extend for every action:* $s \equiv_A t = \max_{a \in A} |v_a(s) - v_a(t)|\,.$

We define a fixed point operator such that it takes in input a distance and gives as output a new distance with determinate properties.

**Definition 5.12 (Distance iterator $F$).** *Given a probabilistic system* $(Q, A, Tr)$*, states* $s, t \in Q$*, an action* $a \in A$*, a probabilistic distributions* $\mu, \nu \in$*Dist*$(Q)$*, a probabilistic pseudometric* $d \colon Q \times Q \to$ *Disc*$(Q)$*, we define the operator between metrics* $F\colon$ *Disc*$(Q) \to$*Disc*$(Q)$ *as*

$$F\left(d\left(s, t\right)\right) = \max\left\{ s \equiv_A t\,,\ \sup_{\mu \in (s \xrightarrow{a} \mu)} \ \inf_{\nu \in (t \xrightarrow{a} \nu)} d(\mu, \nu) \right\} \tag{5.8}$$

*where* $d(\mu, \nu)$ *is a distance defined between the probabilistic distributions generated by the transitions* $s \xrightarrow{a} \mu$ *and* $t \xrightarrow{a} \nu$*.*

The element $s \equiv_a t$ considers the set of actions of both the states. If $s, t$ do not support the same action, i.e. there exists $a \in A$ such that $s \xrightarrow{a} \mu$ and $t \overset{a}{\not\to}$, then we have $|v_a(s) - v_a(t)| = |1 - 0| = 1$. In this case the resulting pseudometric fixes always the value 1, indeed the value of $d$ is in the interval $[0, 1]$. If the sets of actions of, respectively, $s$ and $t$ are identical, then we have $|v_a(s) - v_a(t)| = |0 - 0| = 0$. This happen also when two states are not bisimilar, for this reason $F$ chooses the maximum between the result of equivalence of actions and the better value of the distance $d$.

In the left hand side of Diagram 5.7 we represent the operator $F$ with input the distance $d_1$, which we calculate on states $(s_1, t_1), (s_2, t_2), \ldots, (s_n, t_n)$, and each iteration by an arrow with label $F$. The domain of the resulting distance coincides with the domain of $d_1$, thus $F(d_1(s_i, t_i)) = d_2(s_i, t_i)$ for $i \in \{1, 2, \ldots, n\}$. For this reason we create a simplify version of the diagram in the right hand side of Diag. 5.7, where we omit the states. The fixed point is the distance $d_j(s, t)$, depicted in blue.

$$\begin{tikzcd}
\begin{array}{c} d_1(s_1,t_1) \\ d_1(s_2,t_2) \\ \cdots \\ d_1(s_n,t_n) \end{array} \arrow[d, "\boldsymbol{F}"] & d_1 \arrow[d, "\boldsymbol{F}"] \\
\begin{array}{c} d_2(s_1,t_1) \\ d_2(s_2,t_2) \\ \cdots \\ d_2(s_n,t_n) \end{array} \arrow[d, "\boldsymbol{F}"] & F(d_1) = d_2 \arrow[d, "\boldsymbol{F}"] \\
\begin{array}{c} d_j(s_1,t_1) \\ d_j(s_2,t_2) \\ \cdots \\ d_j(s_n,t_n) \end{array} & F\left(F\left(d_1\right)\right) = d_j
\end{tikzcd} \tag{D.5.7}$$

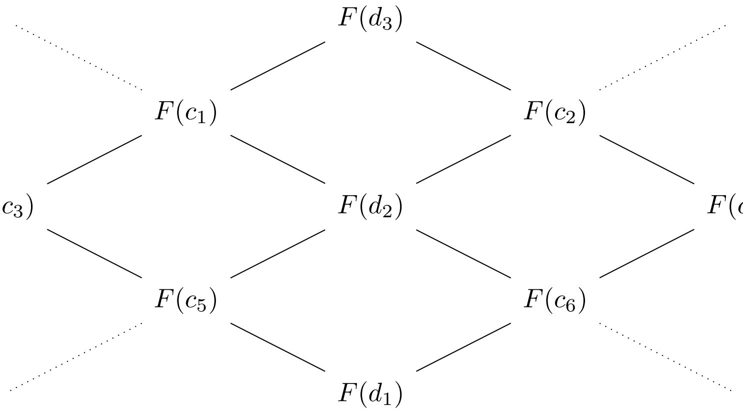
Given several distances on states, we choose a pair of states $(s,t)$ of a probabilistic system and calculate the value of each distance on them. With these values we create an order on the distances, which we represent as a lattice in Diagram 5.8.

$$\begin{tikzcd}
 & & d_3(s,t) = \varepsilon''' & & \\
 & c_1(s,t) = \varepsilon_1 \arrow[ur, dash] \arrow[ul, dotted, dash] & & c_2(s,t) = \varepsilon_2 \arrow[ul, dash] \arrow[ur, dotted, dash] & \\
c_3(s,t) = \varepsilon_3 \arrow[ur, dash] & & d_2(s,t) = \varepsilon'' \arrow[ul, dash] \arrow[ur, dash] & & c_4(s,t) = \varepsilon_4 \arrow[ul, dash] \\
 & c_5(s,t) = \eta_5 \arrow[ul, dash] \arrow[ur, dash] \arrow[dl, dotted, dash] & & c_6(s,t) = \eta_6 \arrow[ul, dash] \arrow[ur, dash] \arrow[dr, dotted, dash] & \\
 & & d_1(s,t) = \varepsilon' \arrow[ul, dash] \arrow[ur, dash] & &
\end{tikzcd} \tag{D.5.8}$$

Defined an order on distances we show that the operator $F$ has an useful property: it is monotone.

**Theorem 5.13 (F monotone).** *Given a probabilistic system $(Q, A, Tr)$ the transformer $F$ is monotonic, that is for each pair of pseudometrics $d_1, d_2 \in [0,1]$ if $d_1 \leqslant d_2$ implies $F(d_1) \leqslant F(d_2)$.*

Now we consider Diag. 5.5 and apply the operator $F$ to each distance. Since $F$ is monotone (Th.5.13), $F$ preserves the order on distances and we represent them in Diagram 5.9. In particular we consider the distances $d_1$ and $d_2$ such that $d_1 \leqslant d_2$, $F$ preserves the order whereby $F(d_1) \leqslant F(d_2)$.



(D.5.9)

In the following we give the proof of the theorem.

*Proof.* We show the monotony with a pointwise comparison assuming that the domain of $d_1$ is the same for $d_2$. By hypothesis we have $d_1 \leqslant d_2$, i.e. for each pair of states $(s,t)$ of the domain $Q$ we have

$$d_1(s,t) \leqslant d_2(s,t).$$

We apply the infimum operator on distribution of $t$ to both the distances, since the outcomes of each pseudometric is a monotone sequence of real numbers belonging to the interval $[0,1]$. Since the first distance is less than the second one in each pair of states, then we have

$$\inf_{\nu \in (t \xrightarrow{a} \nu)} d_1(s,t) \leqslant \inf_{\nu \in (t \xrightarrow{a} \nu)} d_2(s,t). \tag{5.12}$$

We apply the supremum operator on distribution of $s$ to the previous formula, since the above considerations are still verifies. We obtain

$$\sup_{\mu \in (s \xrightarrow{a} \mu)} \inf_{\nu \in (t \xrightarrow{a} \nu)} d_1(s,t) \leqslant \sup_{\mu \in (s \xrightarrow{a} \mu)} \inf_{\nu \in (t \xrightarrow{a} \nu)} d_2(s,t), \tag{5.13}$$

where each element at left hand side of inequality 5.12 has value less or equal to the right hand side of this inequality.

We consider the equivalence of actions $s \equiv_A t$, it is independent from any distances. For pair of states we have

$$\max\left\{s \equiv_A t\,,\ \sup_{\mu \in (s \xrightarrow{a} \mu)} \inf_{\nu \in (t \xrightarrow{a} \nu)} d_1(\mu,\nu)\right\} \leqslant \max\left\{s \equiv_A t\,,\ \sup_{\mu \in (s \xrightarrow{a} \mu)} \inf_{\nu \in (t \xrightarrow{a} \nu)} d_2(\mu,\nu)\right\}, \tag{5.14}$$

since the value of the formula at left hand side of 5.13 is less than the value of the formula at right hand side of this inequality. For Def. 5.12 and for inequality 5.14 we obtain

$$F(d_1(s,t)) \leqslant F(d_2(s,t)).$$

□

In the following we show that $F$ has a fixed point.

**Theorem 5.14.** *Let $(Q, A, Tr)$ be a probabilistic system, PMetric a set of monotone pseudometrics on $Q$ with value interval $[0,1]$, $\leqslant$ be a partial order on pointwise distances, and $d \in Dist(Q)$ be a probabilistic pseudometric on distributions generated on $Q$.*

1. *$(PMetric, \leqslant)$ is a complete lattice,*
2. *$F$ has the least fixed point (lfp) defined by*

$$lfp(F) = \bigcap \{\, d(x,y) \mid F\,(d\,(x,y)) \leqslant d(x,y) \,\}$$

*Proof.*

1. We consider a subset $S \subseteq$ PMetric, we construct the infimum, i.e. a greatest lower bound in (PMetric, $\leqslant$). Let is $d_i \in S$. Since it is defined monotone on all the state in $Q$, we have that for each $s, t, s', t' \in Q$ with a partial order $\leqslant$ if $s \leqslant s'$ and $t \leqslant t'$, then we have $d_i(s,t) \leqslant d_i(s',t')$. We call lower bound of $S$ the distance $d' \in S$ such that $d'(s,t) \leqslant d_i(s,t)$ for each $d_i \in S$ and for each $s, t \in Q$. Since $Q$ is finite and and each $d_i \in S$ is monotone on all the state in $Q$, then the inequality is always defined and the element $d'$ exists. We consider $S' = \{\, d' \in S \mid d' \text{ is a lower bound of } S \,\}$, let $d \in S'$ such that for each $d' \in S'$ and for each $s, t \in Q$ we have $d(s,t) \leqslant d'(s,t)$. It is defined since $Q$ is finite and and each $d_i \in S$ is monotone. The element $d$ is the infimum of $S$.
   A similar construction shows the supremum of $S$, i.e. a least upper bound in (PMetric, $\leqslant$).
   Since every subset $S$ has both infimum and supremum, the partially ordered set (PMetric, $\leqslant$) is a complete lattice.
2. Since (PMetric, $\leqslant$) is a complete lattice and $F$ is monotonous, then for Knaster-Tarski's theorem 3.35 $lfp(F)$ is the least fixed point of $F$.

□

**Definition 5.15.** *The probabilistic simulation metric $\lessdot$ is the least fixpoint (lfp) of $F$.*

Given a relation $R$ with Def. 5.7 we construct the distance $d_L$. Now we consider the distance $d_R$ and we apply the operator $F$. The evaluation of the first step of $F$ on $d_R$ is equivalent to $d_{L,R}$.

**Lemma 5.16.** *Given a probabilistic system $(Q, A, Tr)$, a bisimulation $R$, and $d_R$ the distance such that $d_R(s,t) = 0$ if $s\,R\,t$ and $d_R(s,t) = 1$ otherwise, we have*

$$F(d_R) \;=\; d_{L,R}.$$

*Proof.* Given two states $s, t \in Q$ and the bisimulation $R$, we have two cases.

$(s,t) \in R$: in this case there is no approximation on the simulation. From the transitions $s \xrightarrow{a} \mu$ and $t \xrightarrow{a} \nu$, we can generate a weighting function $w\colon Q \times Q \to [0,1]$ such that $\sum_{s\in Q} w(s,t) = \nu(t)$, $\sum_{t\in Q} w(s,t) = \mu(t)$ and $w(s,t) > 0$. Thus for Def. 3.24 we have $\mu\, L(R)\, \nu$, that is equivalent to $\mu\, L(R,0)\, \nu$. In general for each $s \xrightarrow{a} \mu$ there exists $t \xrightarrow{a} \nu$ such that $\mu\, L(R,0)\, \nu$. For Def. 5.2 we have $s \overset{0}{\sim}_R t$, thus $d_{L,R} = 0$ by Def. 5.7.
For Def. 3.13 $d_R(s,t) = 0$, since $F(d_R)$ is a monotonically decreasing function, i.e. $F(d_R) \leqslant d_R$, then $F(d_R(s,t)) = 0$.

$(s,t) \notin R$: We split the problem in two steps:

1.$F(d_R) \geqslant d_{L,R}$. Since $(s,t) \notin R$, then for definition $d_R(s,t) = 1$. The operator $F$ is a monotonically decreasing function, thus we have $F(d_R) \leqslant d_R$ and $F(d_R) \leqslant 1$.
Since $(s,t) \notin R$ we consider $\mu, \nu$ such that $s \xrightarrow{a} \mu, t \xrightarrow{a} \nu \in Tr$ and a variance $\varepsilon$ such that we generate the decompositions $\mu = \mu_1(1-\varepsilon) + \mu_2\varepsilon$ and $\nu = \nu_1(1-\varepsilon) + \nu_2\varepsilon$. If these decompositions do not exist, then for Def. 5.11 $(s \equiv_A t) = 1$. Otherwise $\mu\, L(R,\varepsilon)\, \nu$, we consider for each $\mu \in s \xrightarrow{a} \mu$ there exists $\nu \in t \xrightarrow{a} \nu$ such that $\mu\, L(R,\varepsilon)\, \nu$ and obtain $\inf_\varepsilon s \overset{\varepsilon}{\sim}_R t$ for Def. 5.7. We replace the formulation "for each" with sup and "there exists" with inf, we consider the infimum $\varepsilon$ and obtain

$$\inf_{\varepsilon\in[0,1]} \sup_{\mu\in s\xrightarrow{a}\mu} \inf_{\nu\in t\xrightarrow{a}\nu} \mu\, L(R,\varepsilon)\, \nu = d_{L,R} \tag{5.15}$$

Formula 5.15 is the definition of $d_{L,R}$. This formula is closed to Formula 5.12 defining $F$, the first part is equal. Then in Formula 5.15 we insert the the choice of minimum $\varepsilon$. Thus $d_{L,R} \leqslant F(d_R)$.
We resume that $d_{L,R} \leqslant 1$ and $F(d_R) \leqslant d_R = 1$, thus $d_{L,R} \leqslant F(d_R)$.

2.$F(d_R) \leqslant d_{L,R}$. We consider the worst case where there exists no action $a$ for $t$ that simulates the action $a$ of $s$. For Def. 5.7 $d_{L,R} = 1$. This case implies that $|v_a(s) - v_a(t)| = 1$, for Def. 5.10 and Def. 5.11, and thus $(s \equiv_A t) = 1$. For Def. 5.12 1 is the maximum value, thus $F(d_R(s,t)) = 1$. This implies that $F(d_R(s,t)) \leqslant d_{L,R}$.

□

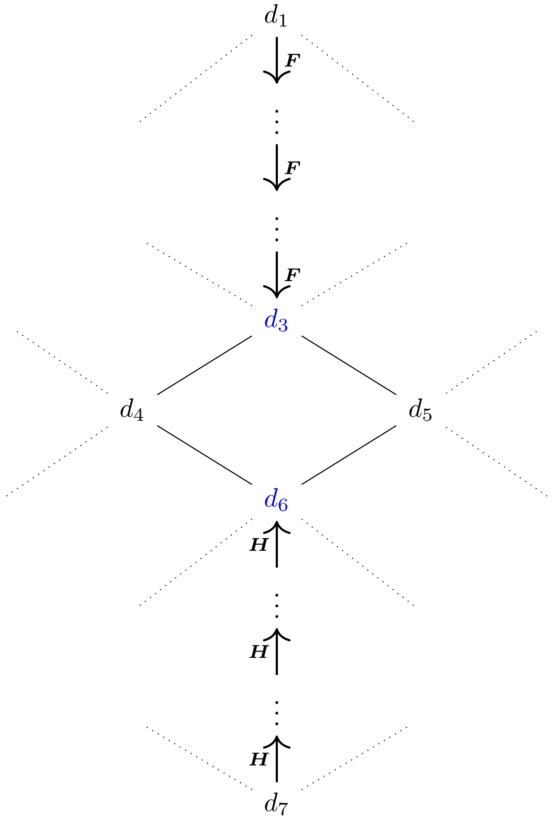
# 6

# Conclusions

In this thesis we have considered probabilistic processes, modelled by automata, to analyses the problem of approximated bisimulations. These relations are used, generally, to simplify the models of some systems and to model agents and attackers in security protocols. In the latter field these relations have been replaced by the metrics, which are the quantitative analogue of probabilistic bisimilarity. Metrics allow a greater precision, since they assign a real number which describes the distance between states or between probabilistic distributions of states. Each one of these distributions describes the probability to enter, from a given state, in a successor state after a transition. Thus a metric is about a degree of similarity between states.

Starting from the notion of $\varepsilon$-(bi)simulation given in [62], we have defined two pseudometrics, called both $d_{L,R}$, on states and on distributions. Our definition of metric between two states is well defined: it is the infimum error $\varepsilon$ such that two given states are still $\varepsilon$-bisimilar, i.e. the probabilistic distributions of these states match except for an error $\varepsilon$ which is the smallest. This formalisation is the main difference with the definitions of metrics in the actual literature. We have considered the largely used Kantorovich metrics, that is the lifting of a given relation $R$ on states which describes the property of the processes to whom we are interested. We have put in relation our metrics with the Kantorovich one. If we consider $d_R$ as starting relation, which is the relation that has value zero on pairs of states in relation and 1 elsewhere, the two metrics are equivalent.

In literature the metrics are operational defined as the fixed point of an iterative transformer. Since this operator is monotone and any set of metrics is a lattice, we can apply Tarski theorem, whose says that a monotonic operator defined on a lattice always has a minimum (or maximum) fixed point. In [12] the operator is a functional transformer $H$ defined on metrics that increases at each iteration. By Tarski theorem the least fixed point is the bisimulation searched. In the thesis we have proposed a recast of this transformer defining a metric transformer called $F$ on probabilistic automata. $F$ and $H$ are similar at high level, the only difference is the definition of the actions: we have labels for the transitions. In our case at each iteration the operator $F$ decreases the values of each pair of states, thus given any metric in input we search the better distance calculating the minimum error error $\varepsilon$ and consider the overlapping (maximum) distance. We have considered the metrics $d_R$ on probabilistic automata, it is a post-fixed-point of $F$. Used as input metric of $F$, after a single iteration the given result is $d_{L,R}$. The value of $F(d_R)$ is less than the initial value of $d_R$, thus the fixed point of $F$ is an over approximation of a metric generated by $H$, which calculates the minimum fixed point as shown in Diagram 6.1. This show also that our operator $F$ is consistent with the literature.



(D.6.1)

The results of this thesis can be expanded to several directions. Using our metrics on automata as basis to study security, the approximate polynomial simulation techniques transposed in metrics is a proof of soundness of logics that generate processes/systems, as simulation techniques are basis for implication languages. This will lead to consider the Turrini's approach as a sound method for verifying distances on automata. A consequent development is investigate the completeness.

Another possible new direction is the extension of our metric $d_{L,R}$ inserting a limit. In details we could fix a limit on the error values, thus the metric will assume only a range of values. The advantage of this consideration is the possibility of analysis of the errors on states directly in a computational level. Until now it has never been possible this approach, the limitations on metrics defined in literature followed a converse approach based on linking a distance with its logical formulation.

As future work we can be use the metrics created and the theorems and the lemma as first step in the analysis of the relation between languages and metrics. A second and interesting future work is the analysis of hemimetrics with simulations, i.e. the pseudometric can be make perfect erasing a constraint and obtaining the definition of hemimetric that better reproduces the asymmetry property of simulation.

# References

1. Alessandro Abate. Approximation Metrics Based on Probabilistic Bisimulations for General State-Space Markov Processes: A Survey. *Electr. Notes Theor. Comput. Sci.*, 297:3–25, 2013.
2. Luca Aceto, Anna Ingòlfsdóttir, Kim G. Larsen, and Jiri Srba. *Reactive Systems: Modelling, Specification and Verification.* Cambridge University Press, August 2007.
3. Jos C. M. Baeten and Davide Sangiorgi. Concurrency Theory: A Historical Perspective on Coinduction and Process Calculi. In Jorg H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 399–442. Elsevier, 2014.
4. Christel Baier and Holger Hermanns. Weak Bisimulation for Fully Probabilistic Processes. In Orna Grumberg, editor, *CAV*, volume 1254 of *Lecture Notes in Computer Science*, pages 119–130. Springer, 1997.
5. Christel Baier, Joost-Pieter Katoen, Holger Hermanns, and Verena Wolf. Comparative branching-time semantics for Markov chains. *Inf. Comput.*, 200(2):149–214, 2005.
6. Hans Bekić. Definable operations in general algebras and the theory of automata and flowcharts. Unpublished manuscript. 1969.
7. Richard Blute, Josee Desharnais, Abbas Edalat, and Prakash Panangaden. Bisimulation for Labelled Markov Processes. In *LICS*, pages 149–158. IEEE Computer Society, 1997.
8. Barry S. Bosik and M. Ümit Uyar. Finite state machine based formal methods in protocol conformance testing: from theory to implementation. *Computer Networks and {ISDN} Systems*, 22(1):7–33, 1991. 9th {IFIP} TC-6 International Symposium on Protocol Specification, Testing and Verification.
9. Carlos A. Cabrelli and Ursula M. Molter. The Kantorovich metric for probability measures on the circle. *Journal of Computational and Applied Mathematics*, 57(3):345–361, 1995.
10. Marek Capinski and Peter E. Kopp. *Measure, integral, and probability.* Springer-Verlag Inc, Berlin; New York, 1999.
11. Liqun Chen. Timed processes: Models, axioms and decidability. Technical report ecs-lfcs-93-271, LFCS, University of Edimgurg, 1993.

12. Luca de Alfaro, Rupak Majumdar, Vishwanath Raman, and Mariëlle Stoelinga. Game Refinement Relations and Metrics. *Logical Methods in Computer Science*, 4(3), 2008.
13. Erik P. de Vink and Jan J. M. M. Rutten. Bisimulation for Probabilistic Transition Systems: A Coalgebraic Approach. *Theor. Comput. Sci.*, 221(1-2):271–293, 1999.
14. Yuxin Deng, Tom Chothia, Catuscia Palamidessi, and Jun Pang. Metrics for Action-labelled Quantitative Transition Systems. *Electr. Notes Theor. Comput. Sci.*, 153(2):79–96, 2006.
15. Yuxin Deng and Wenjie Du. The Kantorovich Metric in Computer Science: A Brief Survey. *Electronic Notes in Theoretical Computer Science*, 253(3):73–82, 2009.
16. Yuxin Deng and Catuscia Palamidessi. Axiomatizations for Probabilistic Finite-State Behaviors. In Vladimiro Sassone, editor, *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2005.
17. Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004.
18. Josee Desharnais, Radha Jagadeesan, Vineet Gupta, and Prakash Panangaden. The Metric Analogue of Weak Bisimulation for Probabilistic Processes. In *LICS*, pages 413–422. IEEE Computer Society, 2002.
19. Josée Desharnais, François Laviolette, and Mathieu Tracol. Approximate Analysis of Probabilistic Processes: Logic, Simulation and Games. In *Fifth International Conference on the Quantitative Evaluaiton of Systems (QEST 2008), 14-17 September 2008, Saint-Malo, France*, pages 264–273. IEEE Computer Society, 2008.
20. Edsger W. Dijkstra. Solution of a Problem in Concurrent Programming Control. *Commun. ACM*, 8(9):569–, September 1965.
21. Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for Finite Markov Decision Processes. In Deborah L. McGuinness and George Ferguson, editors, *AAAI*, pages 950–951. AAAI Press / The MIT Press, 2004.
22. Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for Markov Decision Processes with Infinite State Spaces. In *UAI*, pages 201–208. AUAI Press, 2005.
23. Rainer Gawlick, Roberto Segala, Jørgen Søgaard-Andersen, and Nancy A. Lynch. *Liveness in timed and untimed systems*, pages 166–177. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
24. Alessandro Giacalone, Chi chang Jou, and A. Smolka Scott. Algebraic Reasoning for Probabilistic Concurrent Systems. In *Proc. IFIP TC2 Working Conference on Programming Concepts and Methods*, pages 443–458. North-Holland, 1990.
25. Antoine Girard and George J. Pappas. Approximation Metrics for Discrete and Continuous Systems. *IEEE Trans. Automat. Contr.*, 52(5):782–798, 2007.
26. Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence Journal*, 147:163–224, 2003.
27. Jean Goubault-Larrecq. Une introduction aux capacités, aux jeux et aux prévisions. Le Pavé.
28. Holger Hermanns. *Interactive Markov Chains and the Quest for Quantified Quality.* Springer, 2002.

29. Tony Hoare. Proof of a structured program: 'the sieve of Eratosthenes'. *Comput. J.*, 15(4):321–325, 1972.
30. John E. Hutchinson. Fractals and self-similarity. *Indiana Univ. Math. J.*, 30:713–747, 1981.
31. Bengt Jonsson. *Simulations between specifications of distributed systems*, pages 346–360. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.
32. Bengt Jonsson and Kim Guldstrand Larsen. Specification and Refinement of Probabilistic Processes. In *LICS*, pages 266–277. IEEE Computer Society, 1991.
33. Leonid Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, (Translated in Management Science, 1958):1–4, 1942.
34. Robert M. Keller. Formal Verification of Parallel Programs. *Commun. ACM*, 19(7):371–384, jul 1976.
35. Carlos S. Kubrusly. *Elements of Operator Theory.* Elements of Operator Theory. Birkhäuser Boston, 2001.
36. Marta Z. Kwiatkowska and Gethin Norman. Probabilistic Metric Semantics for a Simple Language with Recursion. In Wojciech Penczek and Andrzej Szalas, editors, *MFCS*, volume 1113 of *Lecture Notes in Computer Science*, pages 419–430. Springer, 1996.
37. Kim Guldstrand Larsen and Arne Skou. Bisimulation through Probabilistic Testing. *Inf. Comput.*, 94(1):1–28, September 1991.
38. Peter Linz. *An introduction to formal languages and automata (4. ed.).* Jones and Bartlett Publishers, 2006.
39. Gavin Lowe. An Attack on the Needham-Schroeder Public-Key Authentication Protocol. *Inf. Process. Lett.*, 56(3):131–133, 1995.
40. Nancy A. Lynch and Mark R. Tuttle. Hierarchical Correctness Proofs for Distributed Algorithms. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '87, pages 137–151, New York, NY, USA, 1987. ACM.
41. Nancy A. Lynch and Frits Vaandrager. *Forward and backward simulations for timing-based systems*, pages 397–446. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
42. Robin Milner. Program simulation: an extended formal notion. *Memo 17, Computer and logic research group*, 1971.
43. Robin Milner. Fully Abstract Models of Typed lambda-Calculi. *Theor. Comput. Sci.*, 4(1):1–22, 1977.
44. Robin Milner. *A Calculus of Communicating Systems.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.
45. Robin Milner. *Communication and Concurrency.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
46. Robin Milner and Mads Tofte. Co-Induction in Relational Semantics. *Theor. Comput. Sci.*, 87(1):209–220, 1991.
47. Mehryar Mohri. Edit-Distance Of Weighted Automata: General Definitions And Algorithms. *Int. J. Found. Comput. Sci.*, 14(6):957–982, 2003.
48. Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.
49. David Park. Fixpoint induction and proofs of program properties. *Machine Intelligence*, 5:59–78, 1969.
50. David Park. *Concurrency and automata on infinite sequences*, pages 167–183. Springer Berlin Heidelberg, Berlin, Heidelberg, 1981.

51. Carl Adam Petri. *Kommunikation mit Automaten.* PhD thesis, Universität Hamburg, 1962.
52. Anna Philippou, Insup Lee, and Oleg Sokolsky. Weak Bisimulation for Probabilistic Systems. In Catuscia Palamidessi, editor, *CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 334–349. Springer, 2000.
53. Gordon D. Plotkin. A Structural Approach to Operational Semantics. Technical report, University of Aarhus, 1981.
54. Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. Environmental Bisimulations for Higher-Order Languages. In *LICS*, pages 293–302. IEEE Computer Society, 2007.
55. Davide Sangiorgi and David Walker. *The Pi-Calculus: a theory of mobile processes.* Cambridge University Press, 2001.
56. Dana Scott and J. de Bakker. A theory of programs. Handwritten notes., 1969.
57. Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1995.
58. Roberto Segala. Modeling and Verification of Randomized Distributed Real -Time Systems. Technical report, Cambridge, MA, USA, 1996.
59. Roberto Segala and Nancy A. Lynch. *Probabilistic simulations for probabilistic processes*, pages 481–496. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
60. Roberto Segala and Nancy A. Lynch. Probabilistic Simulations for Probabilistic Processes. *Nord. J. Comput.*, 2(2):250–273, 1995.
61. Roberto Segala and Andrea Turrini. Approximated Computationally Bounded Simulation Relations for Probabilistic Automata. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium*, CSF '07, pages 140–156, Washington, DC, USA, 2007. IEEE Computer Society.
62. Andrea Turrini. *Hierarchical and Compositional Verification of Cryptographic Protocols.* PhD thesis, University of Verona, 2009.
63. Franck van Breugel, Babita Sharma, and James Worrell. Approximating a Behavioural Pseudometric without Discount for Probabilistic Systems. abs/0803.3796, 2008.
64. Franck van Breugel and James Worrell. An Algorithm for Quantitative Verification of Probabilistic Transition Systems. In Kim Guldstrand Larsen and Mogens Nielsen, editors, *CONCUR*, volume 2154 of *Lecture Notes in Computer Science*, pages 336–350. Springer, 2001.
65. Franck van Breugel and James Worrell. A behavioural pseudometric for probabilistic transition systems. *Theor. Comput. Sci.*, 331(1):115–142, 2005.
66. Leonid N. Vaserstein. Markov Processes over Denumerable Products of Spaces, Describing Large Systems of Automata. *Probl. Peredachi Inf.*, 5(3):64–72, 1969.
67. Jennifer L. Welch, Leslie Lamport, and Nancy A. Lynch. A Lattice-Structured Proof of a Minimum Spanning. In *Proceedings of the Seventh Annual ACM Symposium on Principles of Distributed Computing, Toronto, Ontario, Canada, August 15-17, 1988*, pages 28–43, 1988.
68. Wang Yi. CCS + Time = An Interleaving Model for Real Time Systems. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *ICALP*, volume 510 of *Lecture Notes in Computer Science*, pages 217–228. Springer, 1991.