Behavioural Preorders on Stochastic Systems - Logical, Topological, and Computational Aspects

Pedersen, Mathias Ruggaard

DOI (link to publication from Publisher):
10.54337/aau300041621

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

BEHAVIOURAL PREORDERS ON STOCHASTIC SYSTEMS – LOGICAL, TOPOLOGICAL, AND COMPUTATIONAL ASPECTS

BY
MATHIAS RUGGAARD PEDERSEN

DISSERTATION SUBMITTED 2018

AALBORG UNIVERSITY
DENMARK

Behavioural Preorders on Stochastic Systems - Logical, Topological, and Computational Aspects

Ph.D. Dissertation

Mathias Ruggaard Pedersen

Dissertation submitted October, 2018

Dissertation submitted:	October, 2018
PhD supervisor:	Prof. Dr. Radu Mardare Aalborg University
Assistant PhD supervisor:	Prof. Kim Guldstrand Larsen Aalborg University
PhD committee:	Associate Professor Lisbeth Fajstrup (chairman) Aalborg University
	Associate Professor Marco Carbone IT University of Copenhagen
	Associate Professor Dr. Ana Sokolova Salzburg University
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Computer Science

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-349-5

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2018

Abstract

Computer systems can be found everywhere: in space, in our homes, in our cars, in our pockets, and sometimes even in our own bodies. For concerns of safety, economy, and convenience, it is important that such systems work correctly. However, it is a notoriously difficult task to ensure that the software running on computers behaves correctly and does not contain any bugs.

One approach to ease this task is that of model checking, where a model of the system is made using some mathematical formalism. Requirements expressed in a formal language can then be verified against the model in order to give guarantees that the model satisfies the requirements. If the model is faithful to the system being modelled, then the system itself will also satisfy the requirements.

For many computer systems such as satellites, airbags, and traffic lights, time is an important factor. As such, we need our formalisms and requirement languages to be able to incorporate real time.

In this thesis, we therefore develop formalisms and algorithms that allow us to compare and express properties about real-time systems. We first introduce a logical formalism for reasoning about upper and lower bounds on time, and study the properties of this formalism, including axiomatisation and algorithms for checking when a formula is satisfied.

We then consider the question of when a system is faster than another system. We show that this is a difficult question which can not be answered in general, but we identify some special cases where this question can be answered. We also show that under this notion of faster-than, a local increase in speed may lead to a global decrease in speed. This is known as a timing anomaly, and we take steps toward avoiding such timing anomalies.

Finally, we consider how to compare the real-time behaviour of systems not just qualitatively, but also quantitatively. Thus, we are not just interested in knowing whether a system is faster or slower than another process, but also how much faster or slower it is. This is done by introducing a distance between systems. We show how to compute this distance and that it behaves well with respect to properties expressed in a certain logical formalism.

Resumé

Computersystemer kan findes overalt: i rummet, i vores hjem, i vores biler, i vores lommer og endda nogle gange i vores egne kroppe. På grund af økonomiske, sikkerheds- og bekvemmelighedsmæssige problemstillinger er det vigtigt at sådanne systemer virker korrekt. Det er dog et notorisk svært problem at sørge for at softwaren, der kører på computere, opfører sig korrekt og ikke indeholder fejl.

En tilgang til at gøre dette problem nemmere er modeltestning, hvor man laver en model af systemet i en matematisk formalisme. Derefter kan krav, som er udtrykt i et formelt sprog, blive verificeret op mod modellen for at give en garanti for, at modellen opfylder kravene.

For mange computersystemer såsom satellitter, airbags og trafiklys, er tid en vigtig faktor. På grund af dette har vi behov for formelle sprog til at udtrykke krav og for formalismer, som kan inkorporere realtid.

I denne afhandling udvikler vi derfor formalismer og algoritmer, som lader os sammenligne og udtrykke egenskaber vedrørende realtidssystemer. Først introducerer vi en logisk formalisme til at udtrykke egenskaber om øvre og nedre grænser på tid, og vi studerer denne formalismes egenskaber, herunder aksiomatisering og algoritmer til at fastslå om en formel er opfyldt.

Derefter betragter vi spørgsmålet om, hvornår et system er hurtigere end et andet. Vi viser at dette er et svært spørgsmål, som generelt ikke kan besvares, men vi identificerer nogle specialtilfælde hvor spørgsmålet kan besvares. Vi viser også at med denne opfattelse af hurtigere-end, kan en lokal forøgelse af hastighed føre til et globalt fald i hastighed. Dette kaldes en tidsanomali, og vi påbegynder en undersøgelse af, hvordan sådanne tidsanomalier kan undgås.

Til slut beskæftiger vi os med hvordan vi kan sammenligne systemers realtidsadfærd ikke kun kvalitativt, men også kvantitativt. Dette betyder at vi ikke kun er interesserede i at vide, om et system er hurtigere eller langsommere end et andet system, men også hvor meget hurtigere eller langsommere det er. Dette gør vi ved at introducere en afstand mellem systemer. Vi viser hvordan man kan beregne denne afstand, og vi viser at den kan beskrives ved hjælp af en bestemt logisk formalisme.

Contents

Preface

The research described in this thesis was carried out at Aalborg University from September 2015 to October 2018 as part of the research project Approximate Reasoning for Stochastic Markovian Systems (project number 4181-00360) funded by The Danish Council for Independent Research (DFF-FNU). The aim of this project is to develop an approximation theory for stochastic Markovian systems from a logical, topological, and computational point of view.

The content of this thesis contributes to that aim by developing a notion of a faster-than relation in the context of semi-Markov decision processes, which subsume the popular formalism of continuous-time Markov chains. This allows us to approximate a system by another system which operates slower or faster than the former system. Furthermore, we extend this relation to a distance which can give quantitative information about how closely one system approximates another.

From the logical point of view, we give a logical characterisation of both the faster-than relation and the distance which extends it. Furthermore, we consider common aspects in a logical analysis, such as axiomatisation, satisfiability, and model checking.

From the topological point of view, we consider the topology induced by the distance, and how properties given by a logical specification behave in this topology. In particular, we show that approximate reasoning in the limit is sound, meaning that when approximating closer and closer to the real system, properties enjoyed by the approximations are preserved by the real system.

Finally, from the computational point of view, we develop efficient algorithms for deciding the faster-than relation and for computing the distance.

Acknowledgements

The making of this thesis would not have been possible without the involvement and support of a number of people.

First of all, I would like to thank the Danish Council for Independent Research for funding the DFF-FNU project Approximate Reasoning for Stochastic Markovian Systems which has paid for my PhD studies.

Also thanks to my supervisors Radu Mardare and Kim Guldstrand Larsen for giving me the opportunity to pursue a PhD degree, and for giving me both freedom and support in the process.

Thanks also go to my co-authors Mikkel Hansen, with whom I shared an office and bounced ideas off, Nathanaël Fijalkow, from whom I have learnt much and who kindly hosted me for a week in London, and in particular to Giorgio Bacci, who has been a great help through much of the process.

I thank Christel Baier and Daniel Gburek as well as the rest of the group on algebraic and logical foundations of computer science at TU Dresden for hosting me for three months during my studies.

Special thanks go to Daniel Hillerström, my sparring partner throughout my undergrad studies, with whom I shared stories and worries about the PhD life.

Lastly, but most importantly, I wish to thank my friends and family for their love and support, in particular my parents who I can never thank enough for being the kind of people that I strive to be like, and for always believing in me.

Mathias Ruggaard Pedersen
Aalborg University, October 31, 2018

Part I

Overview

Chapter 1

Introduction

Computer systems today are ubiquitous, from the tiny chips in watches and pacemakers to the massive server farms that power the search engines and other web services that we use every day. For economic, safety, and costumer satisfaction reasons, it is important that such systems function correctly: If a fault is found in a system that has already been mass produced and sold, the manufacturer may have to repair or replace a large number of systems, resulting in additional expenses for the manufacturer. Furthermore, we place high importance on the correctness of safety-critical systems where lives may be at stake, since even small and rare errors may result in serious injury or death, such as the case of the Therac-25 radiation therapy machine, where at least one person died from radiation poisoning due to a software error [100]. Lastly, because computer systems are so common in our everyday lives, we, as customers and users of such systems, have an interest in them working correctly, to save us confusion and frustration.

One of the important aspects of many computer systems is time, especially for *real-time systems* which operate under time constraints. This means that when analysing the correctness of such systems, we want to be able to understand how the system reacts to and evolves over time. Consider the following examples.

Solar-powered satellites: Most satellites orbiting the earth rely on electricity from on-board solar panels. Since such satellites spend periods in the earth's shadow where it can not collect solar power, it is important to correctly plan and schedule the power consumption of the satellite in order to maintain the condition of the battery.

Airbags: The effectiveness of airbags is extremely sensitive to time. Airbags fully inflate in a matter of milliseconds, and if they inflate too soon, they may already have deflated at the moment of impact, whereas if they inflate too late, they may cause more harm than good.

Intelligent traffic lights: Many traffic lights today are equipped with sensors to detect cars arriving at the traffic light. Based on this information, the traffic light can decide how to direct the traffic. Ideally, this could reduce the waiting time for road users in traffic lights.

In all of the above examples, we see that time is an important factor, and numerous such examples exist.

Many of the properties of interest in real-time systems are *non-functional requirements*. Non-functional requirements put constraints on the way in which a system can be realised, in order to make the end user experience more pleasant [34]. Some of the key non-functional requirements that interact with time are reliability, throughput, and response time.

Reliability: A system should function correctly over long periods of time, even as components start to deteriorate.

Throughput: A system should be able to produce or accept output at a consistent and high rate.

Response time: A system should react quickly to inputs given to it.

We therefore need techniques and methods that will allow us to verify properties such as non-functional requirements in real-time systems. One successful approach to verifying the correctness of computer systems is that of *model checking* [35, 56, 117]. The aim of model checking is to build an abstract model of the system in question using some precise mathematical formalism, and then verifying that this model satisfies some specification. Such specifications are often expressed by formulas of some logical formalism, but they can also be represented by another model.

The act of translating a real system into a mathematical formalism, also known as *modeling*, is therefore central to model checking. However, modeling has many difficulties attached to it. One such difficulty is what we will call the *approximate modeling problem*, which is the problem of accurately representing quantitative information such as time in the formalism. All measurements in the real world are made within some error margin, even for highly advanced measuring equipment. The modelling formalism used should therefore be able to accommodate this inaccuracy. Furthermore, many systems have some uncertainty attached to them, such as robots that operate in physical environments or environments where the robot interacts with other agents. Finally, the act of modeling itself requires abstracting away some parts of the real system in order to arrive at a formal model of the essential parts of the system. This abstraction process also introduces an element of error, since elements may be modeled wrongly, or key elements may be left out. It is up to the skill and experience of the person doing the modeling to prevent this from happening.

All of these issues together makes the process of modeling difficult, and we therefore need to develop formalisms, specification languages, and techniques that allow us to account for this uncertainty and approximation. That is the aim of this thesis.

1.1 Structure of Thesis

This thesis is split into two parts. Part I gives an overview of the current state of the art and the papers that are part of this thesis, including the contributions that the papers make to further the state of the art as well as the mathematical preliminaries necessary for the results of the papers.

Part II include the full versions of the papers outlined in Part I. The papers presented in this thesis are extended versions, including detailed proofs and additional material.

Each chapter has its own separate bibliography.

1.2 State of the Art

We first survey the formalisms, specification languages, and techniques that have already been developed for reasoning about real-time systems.

1.2.1 Models

The two most common formalisms for modeling real-time systems are timed automata and Markov chains [5, 18].

Timed automata. Timed automata were introduced by Alur and Dill [4, 5] in the early '90s as a way of modeling time in automata theory. The key concept in timed automata is that of *clocks*, each of which keep track of time and can be independently reset. Transitions can then be constrained such that a transition is only allowed when the current value of each clock satisfies the constraints. Many extensions of timed automata have been considered, many of which include probabilistic behaviour. The most significant, non-probabilistic extension of timed automata is that of timed I/O automata [42, 83], in which an important distinction is made between input and output actions. Probabilistic timed automata [88, 89] are timed automata in which the transitions are given probabilistically, and furthermore, the value to which the clocks reset are also given by a probability distribution. Stochastic timed automata [22] modify the semantics of timed automata, i.e. how the behaviour of a timed automaton is interpreted, rather than modifying the timed automaton itself, which then gives rise to a probabilistic process. There is also another kind of stochastic timed automata, which give different semantics to timed automata. In this semantics, the focus is on many timed

automata operating in a network [41, 80]. Here the semantics is a race between the different components, in the sense that the component that has the smallest delay gets to choose the output.

Markov chains. Markov chains were developed in the early 20th century by Markov and extended to continuous time by Kolmogorov. Nowadays, Markov chains are used in almost all fields of engineering and science. The use of Markov chains in model checking began in the second half of the '80s [40, 98, 132], but probabilistic automata, a generalisation of Markov chains, have been studied in automata theory since their introduction in 1963 by Rabin [118]. Markov chains operate by choosing its transition to the next state according to a probability distribution. In order to model real-time systems, continuous-time Markov chains are often used instead. In continuous-time Markov chains, the transitions are probabilistic as in Markov chains, but in addition the waiting time in each state is governed by an exponential distribution. Various aspects of model checking and specification have been extensively studied for continuous-time Markov chains [10, 17, 19, 32], even for the case of infinite-state chains [69]. However, many phenomena that occur in practice are not exponentially distributed, so it is useful to extend the model of continuous-time Markov chains with distributions other than the exponential. Semi-Markov chains therefore allow the waiting time in a state to be governed by an arbitrary distribution. Aspects of model checking have also been investigated for semi-Markov chains [101], although they have received much less attention in the literature. One can also generalise even further to obtain generalised semi-Markov processes [1, 68] that allow different distributions for different actions. Generalised semi-Markov processes are in fact close in spirit to probabilistic timed automata. They operate by having a set of clocks, one for each possible action. The clocks run down, and when a clock reaches 0, the action associated with that clock is fired. This affects a probabilistic transition to a new state, as well as a probabilistic reset of the clocks according to arbitrary distributions. Another important variant of Markov chains used in model checking is that of Markov decision processes, in which the actions of the process are determined by an outside controller [58, 74]. Finally we mention the model of interactive Markov chains, which combine continuous-time Markov chains with non-deterministic behaviour [76].

Reactive and generative. When discussing probabilistic models, one important distinction is that between reactive and generative models [131]. Reactive systems are those that react to input by taking a transition depending on the input given. On the other hand, generative systems are those that generate output as it executes transitions. As such, reactive systems take inputs, whereas generative systems create outputs. Examples of reactive systems are probabilistic automata, that takes words as input and either accepts or rejects the word, and Markov decision processes, whose behaviour depends on the

input given by the controller. An example of a generative system is that of Segala automata [123], where each state can non-deterministically choose between a number of different generative transitions. Of course, the generative and reactive models can be combined to have both inputs and outputs, as in the model of timed I/O automata.

1.2.2 Logical Specification Languages

Many different logical specification languages have been introduced in the literature for expressing properties related to time.

Weighted logics. For weighted logics, weighted monadic second order logic was introduced to capture the behaviour of weighted automata [51]. This was later extended to many different, closely related models [12, 52, 53, 62, 105]. There have also been attempts to understand the connection between weighted monadic second order logic and probabilistic logics [25]. Weighted modal logic [92] was introduced to reason about the consumption of resources in weighted transition systems. This formalism was later extended to handle recursion [96, 97] and concurrency [94]. A weighted extension of the expressive μ-calculus has also been developed [93].

Timed logics. The two most influential timed logics are linear temporal logic (LTL) [116] and computation tree logic (CTL) [36], both of which are subsumed by CTL* [57], which in turn is subsumed by the μ-calculus [85]. In LTL, time is linear, meaning that at each moment in time, there is only one possible future, whereas in CTL, time is branching, meaning that at each moment in time, we may simultaneously branch out into different future paths. Both LTL and CTL have operators meaning "in the next step, a property will hold", "one property will hold until another property holds", and "a property will eventually hold", as well as many other derived operators. Both LTL and CTL in their original form consider time to be discrete. Therefore real-time extensions have been developed, such as timed CTL [2] for CTL as well as metric interval temporal logic [6] and timed LTL [20, 54] for LTL.

Probabilistic logics. Lastly we discuss logical specification languages for reasoning about probabilistic behaviour. CTL has been extended with probabilistic operators in PCTL [72], where one can express properties such as "with probability at least p, a property will eventually hold". However, like CTL, time is discrete in PCTL. To extend CTL with both probability and real time, continuous stochastic logic (CSL) has been introduced [11, 16, 50]. In another direction, Markovian logic [86, 104] has been introduced, based on earlier work on knowledge and probability [8, 9, 59]. Markovian logic has two kinds of operators, one which states "with at least probability p, we can go to a state satisfying some property", and another which states "with at most probability p, we can go to a state satisfying some property".

1.2.3 Relations Among Computational Processes

There are many ways in which one can compare and relate processes.

Bisimulation. The most popular way to compare processes is that of bisimulation, which was introduced by van Benthem [127] under the name of zigzag connection, and independently by Milner and Park [107, 111] who introduced the name bisimulation. In the context of Markov chains, bisimulation was defined by Larsen and Skou [98] and has close connections to the notion of lumpability [27]. Bisimulation is a notion of behavioural equivalence, in which each process must be able to mimic the behaviour of the other process.

Since the concept of bisimulation has been so successful, notions of bisimulation have been studied for most of the systems that are studied in the literature, including timed automata [3, 29], probabilistic timed automata [125], timed I/O automata [83], Markov chains [46, 82, 98], continuous-time Markov chains [19, 77], continuous-time Markov decision process [109], and generalised semi-Markov processes [68].

Many of these notions of bisimulation follow from a more general theory of processes as coalgebras [46, 119].

Simulation. Instead of asking when processes are behaviourally equivalent, we can ask when one process can simulate or mimic the behaviour of another process. This is the idea behind simulation relations.

Although simulation relations are not as ubiquitous in the literature as bisimulation relations, they have still been studied for many types of systems, including timed automata [126], probabilistic timed automata [125], timed I/O automata [83], Markov chains [47, 81, 128], and continuous-time Markov chains [19].

Simulation has also been studied from the coalgebraic point of view [78].

Trace equivalence and inclusion. Another notion of behavioural equivalence, which comes from automata theory, is that of trace equivalence, in which two processes are said to be equivalent if they have the same possible executions, known as traces. Generally speaking, it has proven more difficult to reason about trace equivalences than to reason about bisimulation [79].

Trace equivalences have been studied for timed automata [3], Markov chains [21], and continuous-time Markov chains [136]

A related, but non-symmetric notion of trace inclusion has also been studied for Markov decision processes [64].

Bisimulation distances. Although the concept of bisimulation has been influential, the concept is not satisfactory for quantitative systems, due to the approximate modeling problem discussed in the introduction. This was originally emphasised by Jou and Smolka [66, 82], who instead of the qualitative bisimulation relations advocated a quantitative bisimulation distance, which not only says when processes behave differently, but also by how much their

behaviour differs.

Such a distance has been developed for timed automata [75] by measuring the difference between the time points along traces of the automata. Distances have also been developed for weighted transition systems [43, 60, 91], including not only bisimulation distances, but also simulation distances that generalise simulation rather than bisimulation.

Much work has been dedicated to studying bisimulation distances for probabilistic systems. One way of defining such a distance is by allowing an approximation factor ε on the probabilities involved in the definition of bisimulation. This leads to the notion of ε-bisimulation, and the distance is then given by the smallest ε that allows for a ε-bisimulation [49, 66]. Another approach is to make use of the Kantorovich[1] distance [45] between probability distributions. This approach has been successfully applied to define bisimulation distances for Markov chains [44, 48, 129], Markov decision processes [61], continuous-time Markov chains [13], and generalised semi-Markov processes [68].

Just as bisimulation distances generalise bisimulation to a quantitative setting, so one could also generalise trace equivalence to a quantitative setting. Together with bisimulation distances, such distances are often called behavioural distances, since they quantify the dissimilarity between the behaviour of systems. Using the total variation distance, such generalisations of trace equivalence has been studied for Markov chains [84] and semi-Markov chains [14], and various other ways to generalise trace equivalence have been considered for non-deterministic Markov chains [28].

Faster-than relations. Another way to relate the behaviour of two processes is to ask when one process is faster than another. For non-probabilistic and discrete-time systems, i.e. systems with no probabilities and where each transition or step takes one time unit, such faster-than relations have been well-studied [39, 102, 103, 108, 120]. In particular, they have been studied for timed automata [67] and for Petri nets [134, 135], which are systems that model the production and consumption of resources as transitions are taken. However, for continuous-time systems, very little work has been done. To the best of our knowledge, the only work that has been done on faster-than relations for continuous-time probabilistic systems is on continuous-time Markov chains [19], where the simulation relation for continuous-time Markov chains contains a condition which informally says that the process which is simulating another process is allowed to fire faster than the process it is simulating.

[1]This distance has many other names, the most notable alternatives being the Wasserstein, Hutchinson, and earth-mover distance.

1.2.4 Algorithms

In order to make use of the concepts we have surveyed so far, one needs algorithms that allow us to determine if e.g. a formula in a logical language can be satisfied or two models are in a certain relation. Preferably, we also want the algorithms to be efficient, so that we can actually run them on computers and get an answer within a reasonable time frame.

Model checking. The model checking problem asks, given a model and a formula, whether the model satisfies that formula. For most logical specification languages, this is a simple problem to verify, and efficient algorithms therefore exist. The model checking problem is in **PTIME** for CTL [36, 37], PCTL [72], and CSL [16]. For LTL [124] and timed CTL [2], the problem is **PSPACE**-complete, whereas it is **EXPSPACE**-complete for metric interval temporal logic [6].

Satisfiability. Another natural problem for a logical language is the satisfiability problem, which asks whether a given formula can be satisfied at all, i.e. whether we can find some model which satisfies the formula. This can be seen as a sanity check for a formula: If a formula is not satisfiable, then it is unreasonable to ask for a system which has the property expressed by the formula. The satisfiability problem is often harder than the model checking problem, since in the model checking problem, we only have to consider a single model, whereas in the satisfiability problem, we have to consider all models. For LTL, the satisfiability problem is **PSPACE**-complete [124, 133], for CTL it is **EXPTIME** [55], for metric interval temporal logic it is **EXPSPACE**-complete [6], and for timed CTL it is undecidable [2]. For PCTL, the decidability (and hence also complexity) of the satisfiability problem is a longstanding open problem, but various fragments have been successfully studied [23, 26, 30, 73, 87, 106]. For a certain weighted logic called recursive weighted logic, the satisfiability problem has been shown to be decidable [95].

Bisimulation and simulation. Since bisimulation, and to some extent simulation, are core concepts when reasoning about the behavior of systems, algorithms have been developed to decide whether two systems are in a simulation or bisimulation relation. Most algorithms for timed automata use the notion of regions, which is a way to partition the state space into finitely many classes. For timed automata, both simulation and bisimulation are **EXPTIME**-complete [29, 90, 99], and the two are in **EXPTIME** as well for probabilistic timed automata [125]. Algorithms for deciding simulation and bisimulation between Markov chains make use of the maximum flow problem for networks [33]. These algorithms are in **PTIME** for Markov chains, continuous-time Markov chains, and Markov decision processes [15, 137].

Trace equivalence and inclusion. Generally speaking, reasoning about traces is much harder than reasoning about bisimulation. For timed au-

tomata, both trace equivalence [5] and trace inclusion [5, 110] are undecidable. See [7] for a survey on these and related results. For probabilistic automata, the trace inclusion problem is also undecidable [24, 38]. However, somewhat surprisingly, the trace equivalence problem for probabilistic automata is in fact decidable and in **PTIME** [64, 122], using techniques from linear algebra. See [63] for an overview of undecidability results for probabilistic automata.

Bisimulation distances. Algorithms to compute bisimulation distances are different in nature from the algorithms we have surveyed so far, since we do not just need a yes/no answer, but must output a number. Therefore it also makes sense to sometimes approximate this number up to an error margin, meaning that instead of computing the number, we compute a number that is sufficiently close to the actual number. For probabilistic systems, computing the bisimulation distance makes use of linear programming to solve the transportation problem (see e.g. [121, pp. 221-223]). This results in a **PTIME**-complete algorithm for Markov chains [31, 130]. An algorithm also exists for continuous-time Markov chains [13], however, technically speaking, this is an approximation algorithm, since the actual value may be irrational. If we instead consider the total variation distance, then the threshold problem, which asks whether the distance is greater than a given threshold, is undecidable [84]. However, the distance can be approximated in **PSPACE** [14, 84].

1.3 Contributions

This section summarises the most significant contributions that this thesis adds to the state of the art. The content of this thesis is based on the following papers.

- Paper A: **Reasoning About Bounds in Weighted Transition Systems** is under submission for Logical Methods in Computer Science [70] and is an extended version of **A Complete Approximation Theory for Weighted Transition Systems**, which was published in the proceedings of the Second International Symposium on Dependable Software Engineering: Theories, Tools, and Applications (2016) [71].
  Co-authors: Mikkel Hansen, Kim Guldstrand Larsen, and Radu Mardare.

- Paper B: **Timed Comparisons of Semi-Markov Processes** was published in the proceedings of the 12th International Conference on Language and Automata Theory and Applications (2018) [115].
  Co-authors: Nathanaël Fijalkow, Giorgio Bacci, Kim Guldstrand Larsen, and Radu Mardare.

- Paper C: **A Faster-Than Relation for Semi-Markov Decision Processes**

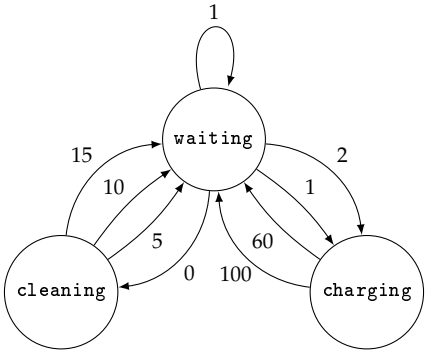


Figure 1.3.1: A simple model of a robot vacuum cleaner.

is based on an unpublished manuscript [113].
Co-authors: Giorgio Bacci and Kim Guldstrand Larsen.

- Paper D: **A Hemimetric Extension of Simulation for Semi-Markov Decision Processes** was published in the proceedings of the 15th International Conference on Quantitative Evaluation of Systems (2018) [114].
  Co-authors: Giorgio Bacci, Kim Guldstrand Larsen, and Radu Mardare.

Paper A takes a simple view of time, where the time that something takes is given explicitly as the weight of taking a transition in a graph. Such systems are called *weighted transition systems*.

Example 1.3.1. Figure 1.3.1 shows an example of a weighted transition system. It is a model of a robot vacuum cleaner which has three states: a `waiting` state, a `cleaning state`, and a `charging` state. In the `waiting` state, the system can choose to simply keep waiting for another minute. However, it can also choose to immediately start cleaning by going to the `cleaning` state. Depending on how dirty the floor is, this cleaning could take 5, 10, or 15 minutes, after which the system returns to the `waiting` state. In the `waiting state`, the system can furthermore decide that it needs to recharge its batteries. This is done by going to the `charging` state, which may take 1 or 2 minutes depending on how far away the robot is from the charging station. The charging itself takes either 60 or 100 minutes, depending on how depleted the batteries are, after which the system returns to the `waiting` state. ⧫

However, because of the approximate modeling problem, putting an exact value for the amount of time that something takes is not feasible. Paper A therefore suggests that we instead reason about lower and upper bounds on the time taken, which is a more manageable engineering task.

This is done by introducing a logical specification language which lets us express properties such as

> "it takes at most 0.1 seconds to go to a state where the airbag is deployed"

and

> "it takes at least 0.01 seconds to go to a state where the airbag is deployed".

The main contributions of Paper A are then to study the properties of this language.

We show first of all that the language describes exactly the behaviour of systems. This means that if two systems have the same behaviour, then they must also satisfy the same properties of our language, and vice versa. This kind of property is known by various names in the literature, including bisimulation invariance, logical characterisation, adequacy, Hennessy-Milner property, and full abstraction.

Contribution 1. We present a language for reasoning about lower and upper bounds in weighted transition systems and we show that this language characterises exactly those systems that have the same kind of behaviour.

We then present a proof system given by a set of axioms which fully describe our logical language, in the sense that anything that can be proved from the axioms must be true, and anything which is true can be proved from the axioms. This means that the axioms are both sound and complete.

Finally, we present two decision algorithms. The first algorithm decides the *model checking problem*, which asks whether a given system satisfies some formula. The second algorithm decides whether, for a given formula in a our language, there exists some weighted transition system in which that formula is true. If such a system exists, the formula is said to be *satisfiable*.

Contribution 2. We provide a complete axiomatisation of the logical specification language, and give an algorithm for deciding the model checking problem and an algorithm for deciding satisfiability of a formula.

Papers B, C, and D take a different view of time. Here, the approximate modeling problem is tackled by introducing probability into the modeling formalism. We thus consider *semi-Markov processes*, where both the time spent in a given state, and the transition step to a next state are governed by probability distributions.

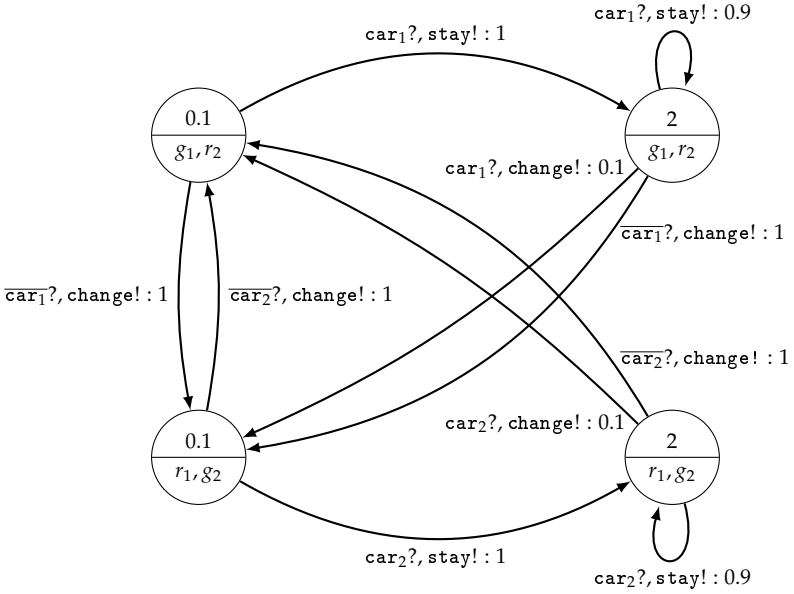


Figure 1.3.2: A simple model of an intelligent traffic light.

Example 1.3.2. Figure 1.3.2 shows an example of a semi-Markov process, modeling a simple traffic light at an intersection with two roads. In this example, there are four states, where the top two states show green light for the first road and red light for the second, and the bottom two states show red light for the first road and green light for the second. This is indicated by the labels g_1 and r_2 in the top two states, and r_1 and g_2 in the bottom two states. The number in each state denotes the rate of an exponential distribution, meaning that the higher the number, the faster that state will take a transition. The labels on the edges between states denote what action is taken, whether it is an input or an output, and the probability of taking that action. Inputs are denoted by ? and outputs by !. For example, $\texttt{car}_2?, \texttt{stay}! : 0.9$ means that when receiving the input $\texttt{car}_2$, the system has a 90% chance to take this transition, which also outputs `stay`.

Imagine that the system starts in the top left state, where the light is green for the first road and red for the second. If now the system sees a car approaching on the first road, indicated by the action $\texttt{car}_1?$, then the system changes to the top right state, in which the system has a high chance of keeping the light green for the first road. Hence, seeing a car approaching on the first road will tend to cause the light to remain green for that road. On the other hand, if a car is observed on the second road, both roads, or none of the roads, all of which is indicated by the action $\overline{\texttt{car}_1}?$, then the system

changes the red and green lights by going to the bottom left state, in which the light is red for the first road and green for the second, and outputting `change`!. The top right state has a higher rate than the top left state, since in this state the system needs to be able to quickly react to a car approaching on the second road, so that the second road is not stuck with red light for too long. In this state, when another car is observed on the first road, again indicated by $\texttt{car}_1$?, then the system has probability 0.9 of staying in this state and outputting `stay`!, thus keeping the light green for the first road a little longer. However, even if another car is observed on the first road, the system may still change the lights with probability 0.1, out of fairness to pedestrians who are not covered by the model. If a car is observed on the second road, both roads, or none of the roads while in this state, the system again changes to the bottom left state, thus changing the light to green for the second road.

The bottom two states function symmetrically, with the light being green for the first road and red for the second road. ⧫

Papers B and C consider how to compare semi-Markov processes with respect to the amount of time it takes to execute sequences of actions. Here, the idea is that one process should be faster than another if whatever sequence of actions the slow process can do, the fast process can do faster and with a higher probability. This is the so-called trace-based semantics of semi-Markov processes, hence we will refer to this as the trace-based faster-than relation.

We show that the trace-based faster-than relation is undecidable, meaning that there is no algorithm which can determine whether a given process is faster than another. This undecidability result is quite robust, since the relation remains undecidable even if we consider approximating the relation up to a multiplicative error term.

Contribution 3. We show that deciding the trace-based faster-than relation is a difficult problem. In particular, the relation is undecidable and approximating it up to a multiplicative constant is impossible.

However, we still obtain some positive results. If we consider approximation up to an additive constant rather than a multiplicative constant, then we can recover decidability under the following two assumptions. The first assumption is that we only consider the behaviour of the processes up to some finite point in time. Thus we allow the fast process to become slower than the other process, as long as this happens only sufficiently far into the future. The second assumption is that a process must spend some non-zero amount of time in each state that it visits. In other words, instantaneous change of state is not allowed, which is a reasonable assumption from a practical point of view. Such processes are called slow.

Contribution 4. We give an algorithm for approximating a time-bounded version of the trace-based faster-than relation up to an additive constant for slow processes.

Another way to recover decidability is to restrict ourselves to so-called unambiguous processes. An unambiguous process is one in which the next state is determined uniquely by the output, so if you know the current state of the process and you know what is output, then you know exactly what state the process ends up in next.

Contribution 5. We give an algorithm for unambiguous processes which can decide whether one process is trace-based faster than another.

We also study the trace-based faster-than relation from the logical point of view. We describe a simple logical language which is expressive enough to characterise exactly those states that are related by the trace-based faster-than relation. We study the properties of the language and show in particular that both the satisfiability and the model checking problem are decidable.

Contribution 6. We introduce a logical language which characterises the trace-based faster-than relation and we show that both the satisfiability problem and the model checking problem for this language are decidable.

Lastly we consider the compositional aspects of the trace-based faster-than relation. When considering a number of components operating in parallel, we would like to replace one or more of these components with another component which is faster. However, we show that doing so may lead to parallel timing anomalies, meaning that although the replaced component is faster than the previous component, this may result in the overall system becoming slower. In an attempt to better understand such parallel timing anomalies and how to avoid them, we identify a set of conditions which will ensure that parallel timing anomalies do not occur. Furthermore, we give an algorithm to check whether these conditions are satisfied.

Contribution 7. We give examples of parallel timing anomalies occuring for the trace-based faster-than relation. However, we also describe some conditions under which parallel timing anomalies can not occur, and we develop an algorithm for checking whether these conditions are met.

Paper D also considers semi-Markov processes. However, here we furthermore address the approximate modeling problem by not only comparing

processes qualitatively, but also quantitatively. Thus we are not only able to say whether a process is faster than another, but are also able to quantify how much slower or faster it is.

For this we consider what we will call the simulation-based faster-than relation, so called because it is based on the idea of one process simulating another. Roughly speaking, a process is simulation-based faster than another process if every step in the slow process can be simulated by the faster process, except the faster process is allowed to do the step faster. To turn this into a quantitative measure, we introduce an acceleration factor through which we obtain a distance between processes. The acceleration factor allows us to increase the speed of a process, so that by accelerating a process by this factor, it may become faster than another process which it was originally slower than. We then define the distance between from one process to another as the smallest acceleration factor necessary to make second process faster than the first. Our first result is an algorithm for computing this distance.

Contribution 8. We describe an algorithm for computing the distance from one process to another. This algorithm runs in polynomial time using known techniques, making it relevant for use and implementation in practice.

We then consider the compositional aspects of the distance. Here the relevant notion is that of non-expansiveness: Composition should not expand the distance between processes. If composition is non-expansive, then it also follows that we are guaranteed that parallel timing anomalies do not occur. We show that under some mild conditions, which are satisfied by many types of composition found in the literature, composition is indeed non-expansive with respect to our distance.

Contribution 9. We show that, under mild assumptions, composition is non-expansive with respect to the distance between semi-Markov processes.

Finally we consider a logical language which we call timed Markovian logic. This language can express properties such as

> "with probability at least 0.99 we leave the state where the traffic light is red before 10 seconds have passed"

and

> "with probability at most 0.5 we end up in a state where the traffic light is green."

We show that this language gives a logical characterisation of the simulation-based faster-than relation. Furthermore, we extend this to a quantitative generalisation of logical characterisation for our distance.

Contribution 10. We introduce a logical specification language called timed Markovian logic and show that this language characterises both the simulation-based faster-than relation and the distance between semi-Markov processes.

The rest of Part I is structured as follows. In Chapter 2 we introduce some mathematical concepts and notation that we will use throughout the thesis, including the definition of weighted transition systems and semi-Markov processes that we have mentioned here. We describe in more detail the results of Paper A in Chapter 3, the results of Paper B and Paper C in Chapter 4, and the results of Paper D in Chapter 5. For the complete details, see the full papers in Part II.

1.4 References

[1] R. Alur and M. Bernadsky, "Bounded model checking for GSMP models of stochastic real-time systems," in *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings*, ser. Lecture Notes in Computer Science, J. P. Hespanha and A. Tiwari, Eds., vol. 3927. Springer, 2006, pp. 19–33. [Online]. Available: https://doi.org/10.1007/11730637_5

[2] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking in dense real-time," *Inf. Comput.*, vol. 104, no. 1, pp. 2–34, 1993. [Online]. Available: https://doi.org/10.1006/inco.1993.1024

[3] R. Alur, C. Courcoubetis, and T. A. Henzinger, "The observational power of clocks," in *CONCUR '94, Concurrency Theory, 5th International Conference, Uppsala, Sweden, August 22-25, 1994, Proceedings*, ser. Lecture Notes in Computer Science, B. Jonsson and J. Parrow, Eds., vol. 836. Springer, 1994, pp. 162–177. [Online]. Available: https://doi.org/10.1007/978-3-540-48654-1_16

[4] R. Alur and D. L. Dill, "Automata for modeling real-time systems," in *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, July 16-20, 1990, Proceedings*, ser. Lecture Notes in Computer Science, M. Paterson, Ed., vol. 443. Springer, 1990, pp. 322–335. [Online]. Available: https://doi.org/10.1007/BFb0032042

[5] ——, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994. [Online]. Available: https://doi.org/10.1016/0304-3975(94)90010-8

[6] R. Alur, T. Feder, and T. A. Henzinger, "The benefits of relaxing punctuality," *J. ACM*, vol. 43, no. 1, pp. 116–146, 1996. [Online]. Available: http://doi.acm.org/10.1145/227595.227602

[7] R. Alur and P. Madhusudan, "Decision problems for timed automata: A survey," in *Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM-RT 2004, Bertinoro, Italy, September 13-18, 2004, Revised Lectures*, ser. Lecture Notes in Computer Science, M. Bernardo and F. Corradini, Eds., vol. 3185. Springer, 2004, pp. 1–24. [Online]. Available: https://doi.org/10.1007/978-3-540-30080-9_1

[8] R. J. Aumann, "Interactive epistemology I: knowledge," *Int. J. Game Theory*, vol. 28, no. 3, pp. 263–300, 1999. [Online]. Available: https://doi.org/10.1007/s001820050111

[9] ——, "Interactive epistemology II: probability," *Int. J. Game Theory*, vol. 28, no. 3, pp. 301–314, 1999. [Online]. Available: https://doi.org/10.1007/s001820050112

[10] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Model-checking continuous-time Markov chains," *ACM Trans. Comput. Logic*, vol. 1, no. 1, pp. 162–170, Jul. 2000. [Online]. Available: http://doi.acm.org/10.1145/343369.343402

[11] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton, "Verifying continuous time Markov chains," in *Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings*, ser. Lecture Notes in Computer Science, R. Alur and T. A. Henzinger, Eds., vol. 1102. Springer, 1996, pp. 269–276. [Online]. Available: https://doi.org/10.1007/3-540-61474-5_75

[12] P. Babari, M. Droste, and V. Perevoshchikov, "Weighted register automata and weighted logic on data words," in *Theoretical Aspects of Computing - ICTAC 2016 - 13th International Colloquium, Taipei, Taiwan, ROC, October 24-31, 2016, Proceedings*, ser. Lecture Notes in Computer Science, A. Sampaio and F. Wang, Eds., vol. 9965, 2016, pp. 370–384. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46750-4_21

[13] G. Bacci, G. Bacci, K. G. Larsen, and R. Mardare, "On-the-fly computation of bisimilarity distances," *Logical Methods in Computer Science*, vol. 13, no. 2, 2017. [Online]. Available: https://doi.org/10.23638/LMCS-13(2:13)2017

[14] ——, "On the total variation distance of semi-Markov chains," in *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, ser. Lecture Notes in Computer Science, A. M. Pitts, Ed., vol. 9034. Springer, 2015, pp. 185–199. [Online]. Available: https://doi.org/10.1007/978-3-662-46678-0_12

[15] C. Baier, B. Engelen, and M. E. Majster-Cederbaum, "Deciding bisimilarity and similarity for probabilistic processes," *J. Comput. Syst. Sci.*, vol. 60, no. 1, pp. 187–231, 2000. [Online]. Available: http://dx.doi.org/10.1006/jcss.1999.1683

[16] C. Baier, B. R. Haverkort, H. Hermanns, and J. Katoen, "Model checking continuous-time Markov chains by transient analysis," in *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, ser. Lecture Notes in Computer Science, E. A. Emerson and A. P. Sistla, Eds., vol. 1855. Springer, 2000, pp. 358–372. [Online]. Available: https://doi.org/10.1007/10722167_28

[17] ——, "Model-checking algorithms for continuous-time Markov chains," *IEEE Trans. Software Eng.*, vol. 29, no. 6, pp. 524–541, 2003. [Online]. Available: https://doi.org/10.1109/TSE.2003.1205180

[18] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[19] C. Baier, J. Katoen, H. Hermanns, and V. Wolf, "Comparative branching-time semantics for Markov chains," *Inf. Comput.*, vol. 200, no. 2, pp. 149–214, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2005.03.001

[20] A. Bauer, M. Leucker, and C. Schallhart, "Monitoring of real-time properties," in *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, ser. Lecture Notes in Computer Science, S. Arun-Kumar and N. Garg, Eds., vol. 4337. Springer, 2006, pp. 260–272. [Online]. Available: https://doi.org/10.1007/11944836_25

[21] M. Bernardo, "Markovian testing and trace equivalences exactly lump more than Markovian bisimilarity," *Electr. Notes Theor. Comput. Sci.*, vol. 162, pp. 87–99, 2006. [Online]. Available: https://doi.org/10.1016/j.entcs.2005.12.079

[22] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Größer, and M. Jurdzinski, "Stochastic timed automata," *Logical Methods in Computer Science*, vol. 10, no. 4, 2014. [Online]. Available: https://doi.org/10.2168/LMCS-10(4:6)2014

[23] N. Bertrand, J. Fearnley, and S. Schewe, "Bounded satisfiability for PCTL," in *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France*, ser. LIPIcs, P. Cégielski and A. Durand, Eds., vol. 16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012, pp. 92–106. [Online]. Available: https://doi.org/10.4230/LIPIcs.CSL.2012.92

[24] V. D. Blondel and V. Canterini, "Undecidable problems for probabilistic automata of fixed dimension," *Theory Comput. Syst.*, vol. 36, no. 3, pp. 231–245, 2003.

[25] B. Bollig and P. Gastin, "Weighted versus probabilistic logics," in *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, ser. Lecture Notes in Computer Science, V. Diekert and D. Nowotka, Eds., vol. 5583. Springer, 2009, pp. 18–38. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02737-6_2

[26] T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera, "The satisfiability problem for probabilistic CTL," in *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24-27 June 2008, Pittsburgh, PA, USA*. IEEE Computer Society, 2008, pp. 391–402. [Online]. Available: https://doi.org/10.1109/LICS.2008.21

[27] P. Buchholz, "Exact and ordinary lumpability in finite Markov chains," *Journal of Applied Probability*, vol. 31, no. 1, pp. 59–75, 1994. [Online]. Available: http://www.jstor.org/stable/3215235

[28] V. Castiglioni, "Trace and testing metrics on nondeterministic probabilistic processes," in *Proceedings Combined 25th International Workshop on Expressiveness*

in Concurrency and 15th Workshop on Structural Operational Semantics and 15th Workshop on Structural Operational Semantics, EXPRESS/SOS 2018, Beijing, China, September 3, 2018., ser. EPTCS, J. A. Pérez and S. Tini, Eds., vol. 276, 2018, pp. 19–36. [Online]. Available: https://doi.org/10.4204/EPTCS.276.4

[29] K. Cerans, "Decidability of bisimulation equivalences for parallel timer processes," in *Computer Aided Verification, Fourth International Workshop, CAV '92, Montreal, Canada, June 29 - July 1, 1992, Proceedings*, ser. Lecture Notes in Computer Science, G. von Bochmann and D. K. Probst, Eds., vol. 663. Springer, 1992, pp. 302–315. [Online]. Available: https://doi.org/10.1007/3-540-56496-9_24

[30] S. Chakraborty and J. Katoen, "On the satisfiability of some simple probabilistic logics," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, M. Grohe, E. Koskinen, and N. Shankar, Eds. ACM, 2016, pp. 56–65. [Online]. Available: http://doi.acm.org/10.1145/2933575.2934526

[31] D. Chen, F. van Breugel, and J. Worrell, "On the complexity of computing probabilistic bisimilarity," in *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, ser. Lecture Notes in Computer Science, L. Birkedal, Ed., vol. 7213. Springer, 2012, pp. 437–451. [Online]. Available: https://doi.org/10.1007/978-3-642-28729-9_29

[32] T. Chen, T. Han, J. Katoen, and A. Mereacre, "Model checking of continuous-time Markov chains against timed automata specifications," *Logical Methods in Computer Science*, vol. 7, no. 1, 2011. [Online]. Available: https://doi.org/10.2168/LMCS-7(1:12)2011

[33] J. Cheriyan, T. Hagerup, and K. Mehlhorn, "Can a maximum flow be computed on o(nm) time?" in *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, July 16-20, 1990, Proceedings*, ser. Lecture Notes in Computer Science, M. Paterson, Ed., vol. 443. Springer, 1990, pp. 235–248. [Online]. Available: https://doi.org/10.1007/BFb0032035

[34] L. Chung and J. C. S. do Prado Leite, "On non-functional requirements in software engineering," in *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, ser. Lecture Notes in Computer Science, A. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. K. Yu, Eds., vol. 5600. Springer, 2009, pp. 363–379. [Online]. Available: https://doi.org/10.1007/978-3-642-02463-4_19

[35] E. M. Clarke, "The birth of model checking," in *25 Years of Model Checking - History, Achievements, Perspectives*, ser. Lecture Notes in Computer Science, O. Grumberg and H. Veith, Eds., vol. 5000. Springer, 2008, pp. 1–26. [Online]. Available: https://doi.org/10.1007/978-3-540-69850-0_1

[36] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching-time temporal logic," in *Logics of Programs, Workshop, Yorktown Heights, New York, May 1981*, ser. Lecture Notes in Computer Science,

D. Kozen, Ed., vol. 131. Springer, 1981, pp. 52–71. [Online]. Available: https://doi.org/10.1007/BFb0025774

[37] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Trans. Program. Lang. Syst.*, vol. 8, no. 2, pp. 244–263, 1986. [Online]. Available: http://doi.acm.org/10.1145/5397.5399

[38] A. Condon and R. J. Lipton, "On the complexity of space bounded interactive proofs (extended abstract)," in *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*. IEEE Computer Society, 1989, pp. 462–467. [Online]. Available: https://doi.org/10.1109/SFCS.1989.63519

[39] F. Corradini, R. Gorrieri, and M. Roccetti, "Performance preorder: Ordering processes with respect to speed," in *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, ser. Lecture Notes in Computer Science, J. Wiedermann and P. Hájek, Eds., vol. 969. Springer, 1995, pp. 444–453. [Online]. Available: https://doi.org/10.1007/3-540-60246-1_150

[40] C. Courcoubetis and M. Yannakakis, "Verifying temporal properties of finite-state probabilistic programs," in *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*. IEEE Computer Society, 1988, pp. 338–345. [Online]. Available: https://doi.org/10.1109/SFCS.1988.21950

[41] A. David, K. G. Larsen, A. Legay, M. Mikucionis, D. B. Poulsen, J. van Vliet, and Z. Wang, "Statistical model checking for networks of priced timed automata," in *Formal Modeling and Analysis of Timed Systems - 9th International Conference, FORMATS 2011, Aalborg, Denmark, September 21-23, 2011. Proceedings*, ser. Lecture Notes in Computer Science, U. Fahrenberg and S. Tripakis, Eds., vol. 6919. Springer, 2011, pp. 80–96. [Online]. Available: https://doi.org/10.1007/978-3-642-24310-3_7

[42] A. David, K. G. Larsen, A. Legay, U. Nyman, and A. Wasowski, "Timed I/O automata: a complete specification theory for real-time systems," in *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*, K. H. Johansson and W. Yi, Eds. ACM, 2010, pp. 91–100. [Online]. Available: http://doi.acm.org/10.1145/1755952.1755967

[43] L. de Alfaro, M. Faella, and M. Stoelinga, "Linear and branching system metrics," *IEEE Trans. Software Eng.*, vol. 35, no. 2, pp. 258–273, 2009. [Online]. Available: https://doi.org/10.1109/TSE.2008.106

[44] Y. Deng, T. Chothia, C. Palamidessi, and J. Pang, "Metrics for action-labelled quantitative transition systems," *Electr. Notes Theor. Comput. Sci.*, vol. 153, no. 2, pp. 79–96, 2006. [Online]. Available: https://doi.org/10.1016/j.entcs.2005.10.033

[45] Y. Deng and W. Du, "The kantorovich metric in computer science: A brief survey," *Electr. Notes Theor. Comput. Sci.*, vol. 253, no. 3, pp. 73–82, 2009. [Online]. Available: https://doi.org/10.1016/j.entcs.2009.10.006

[46] J. Desharnais, A. Edalat, and P. Panangaden, "Bisimulation for labelled Markov processes," *Inf. Comput.*, vol. 179, no. 2, pp. 163–193, 2002. [Online]. Available: https://doi.org/10.1006/inco.2001.2962

[47] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Approximating labelled Markov processes," *Inf. Comput.*, vol. 184, no. 1, pp. 160–200, 2003. [Online]. Available: https://doi.org/10.1016/S0890-5401(03)00051-8

[48] ——, "Metrics for labelled Markov processes," *Theor. Comput. Sci.*, vol. 318, no. 3, pp. 323–354, 2004. [Online]. Available: https://doi.org/10.1016/j.tcs.2003.09.013

[49] J. Desharnais, F. Laviolette, and M. 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*. IEEE Computer Society, 2008, pp. 264–273. [Online]. Available: https://doi.org/10.1109/QEST.2008.42

[50] J. Desharnais and P. Panangaden, "Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes," *J. Log. Algebr. Program.*, vol. 56, no. 1-2, pp. 99–115, 2003. [Online]. Available: https://doi.org/10.1016/S1567-8326(02)00068-1

[51] M. Droste and P. Gastin, "Weighted automata and weighted logics," in *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580. Springer, 2005, pp. 513–525. [Online]. Available: http://dx.doi.org/10.1007/11523468_42

[52] M. Droste and G. Rahonis, "Weighted automata and weighted logics on infinite words," in *Developments in Language Theory, 10th International Conference, DLT 2006, Santa Barbara, CA, USA, June 26-29, 2006, Proceedings*, ser. Lecture Notes in Computer Science, O. H. Ibarra and Z. Dang, Eds., vol. 4036. Springer, 2006, pp. 49–58. [Online]. Available: http://dx.doi.org/10.1007/11779148_6

[53] M. Droste and H. Vogler, "Weighted tree automata and weighted logics," *Theor. Comput. Sci.*, vol. 366, no. 3, pp. 228–247, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2006.08.025

[54] D. D'Souza, "A logical characterisation of event clock automata," *Int. J. Found. Comput. Sci.*, vol. 14, no. 4, pp. 625–640, 2003. [Online]. Available: https://doi.org/10.1142/S0129054103001923

[55] E. A. Emerson, "Temporal and modal logic," in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, 1990, pp. 995–1072.

[56] E. A. Emerson and E. M. Clarke, "Characterizing correctness properties of parallel programs using fixpoints," in *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherland, July 14-18, 1980, Proceedings*, ser. Lecture Notes in Computer Science, J. W. de Bakker and J. van Leeuwen, Eds., vol. 85. Springer, 1980, pp. 169–181. [Online]. Available: https://doi.org/10.1007/3-540-10003-2_69

[57] E. A. Emerson and J. Y. Halpern, ""sometimes" and "not never" revisited: on branching versus linear time temporal logic," *J. ACM*, vol. 33, no. 1, pp. 151–178, 1986. [Online]. Available: http://doi.acm.org/10.1145/4904.4999

[58] K. Etessami, M. Z. Kwiatkowska, M. Y. Vardi, and M. Yannakakis, "Multi-objective model checking of Markov decision processes," *Logical Methods in Computer Science*, vol. 4, no. 4, 2008. [Online]. Available: https://doi.org/10.2168/LMCS-4(4:8)2008

[59] R. Fagin and J. Y. Halpern, "Reasoning about knowledge and probability," *J. ACM*, vol. 41, no. 2, pp. 340–367, 1994. [Online]. Available: http://doi.acm.org/10.1145/174652.174658

[60] U. Fahrenberg, C. R. Thrane, and K. G. Larsen, "Distances for weighted transition systems: Games and properties," in *Proceedings Ninth Workshop on Quantitative Aspects of Programming Languages, QAPL 2011, Saarbrücken, Germany, April 1-3, 2011.*, ser. EPTCS, M. Massink and G. Norman, Eds., vol. 57, 2011, pp. 134–147. [Online]. Available: https://doi.org/10.4204/EPTCS.57.10

[61] N. Ferns, P. Panangaden, and D. Precup, "Bisimulation metrics for continuous Markov decision processes," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1662–1714, 2011. [Online]. Available: https://doi.org/10.1137/10080484X

[62] I. Fichtner, "Weighted picture automata and weighted logics," *Theory Comput. Syst.*, vol. 48, no. 1, pp. 48–78, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00224-009-9225-3

[63] N. Fijalkow, "Undecidability results for probabilistic automata," *SIGLOG News*, vol. 4, no. 4, pp. 10–17, 2017. [Online]. Available: http://doi.acm.org/10.1145/3157831.3157833

[64] N. Fijalkow, S. Kiefer, and M. Shirmohammadi, "Trace refinement in labelled Markov decision processes," in *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, ser. Lecture Notes in Computer Science, B. Jacobs and C. Löding, Eds., vol. 9634. Springer, 2016, pp. 303–318. [Online]. Available: https://doi.org/10.1007/978-3-662-49630-5_18

[65] M. Fränzle, D. Kapur, and N. Zhan, Eds., *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016, Beijing, China, November 9-11, 2016, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9984, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-47677-3

[66] A. Giacalone, C.-C. Jou, and S. A. Smolka, "Algebraic reasoning for probabilistic concurrent systems," in *Proc. IFIP TC2 Working Conference on Programming Concepts and Methods*. North-Holland, 1990, pp. 443–458.

[67] S. Guha, C. Narayan, and S. Arun-Kumar, "On decidability of prebisimulation for timed automata," in *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings*, ser. Lecture Notes in Computer Science, P. Madhusudan and S. A.

Seshia, Eds., vol. 7358. Springer, 2012, pp. 444–461. [Online]. Available: https://doi.org/10.1007/978-3-642-31424-7_33

[68] V. Gupta, R. Jagadeesan, and P. Panangaden, "Approximate reasoning for real-time probabilistic processes," *Logical Methods in Computer Science*, vol. 2, no. 1, 2006. [Online]. Available: https://doi.org/10.2168/LMCS-2(1:4)2006

[69] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang, "Time-bounded model checking of infinite-state continuous-time Markov chains," *Fundam. Inform.*, vol. 95, no. 1, pp. 129–155, 2009. [Online]. Available: https://doi.org/10.3233/FI-2009-145

[70] M. Hansen, K. G. Larsen, R. Mardare, and M. R. Pedersen, "Reasoning about bounds in weighted transition systems," *CoRR*, vol. abs/1703.03346, 2017. [Online]. Available: http://arxiv.org/abs/1703.03346

[71] M. Hansen, K. G. Larsen, R. Mardare, M. R. Pedersen, and B. Xue, "A complete approximation theory for weighted transition systems," in *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016, Beijing, China, November 9-11, 2016, Proceedings*, ser. Lecture Notes in Computer Science, M. Fränzle, D. Kapur, and N. Zhan, Eds., vol. 9984, 2016, pp. 213–228. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-47677-3_14

[72] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Asp. Comput.*, vol. 6, no. 5, pp. 512–535, 1994. [Online]. Available: https://doi.org/10.1007/BF01211866

[73] S. Hart and M. Sharir, "Probabilistic propositional temporal logics," *Information and Control*, vol. 70, no. 2/3, pp. 97–155, 1986. [Online]. Available: https://doi.org/10.1016/S0019-9958(86)80001-8

[74] D. Henriques, J. Martins, P. Zuliani, A. Platzer, and E. M. Clarke, "Statistical model checking for Markov decision processes," in *Ninth International Conference on Quantitative Evaluation of Systems, QEST 2012, London, United Kingdom, September 17-20, 2012*. IEEE Computer Society, 2012, pp. 84–93. [Online]. Available: https://doi.org/10.1109/QEST.2012.19

[75] T. A. Henzinger, R. Majumdar, and V. S. Prabhu, "Quantifying similarities between timed systems," in *Formal Modeling and Analysis of Timed Systems, Third International Conference, FORMATS 2005, Uppsala, Sweden, September 26-28, 2005, Proceedings*, ser. Lecture Notes in Computer Science, P. Pettersson and W. Yi, Eds., vol. 3829. Springer, 2005, pp. 226–241. [Online]. Available: https://doi.org/10.1007/11603009_18

[76] H. Hermanns, *Interactive Markov Chains: The Quest for Quantified Quality*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2428. [Online]. Available: https://doi.org/10.1007/3-540-45804-2

[77] J. Hillston, *A compositional approach to performance modelling*, ser. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996. [Online]. Available: https://doi.org/10.1017/CBO9780511569951

[78] J. Hughes and B. Jacobs, "Simulations in coalgebra," *Theor. Comput. Sci.*, vol. 327, no. 1-2, pp. 71–108, 2004. [Online]. Available: https://doi.org/10.1016/j.tcs.2004.07.022

[79] H. Hüttel and S. Shukla, "On the complexity of deciding behavioural equivalences and preorders. a survey," *BRICS Report Series*, vol. 3, no. 39, 1996. [Online]. Available: https://tidsskrift.dk/brics/article/view/20021

[80] C. Jégourel, K. G. Larsen, A. Legay, M. Mikucionis, D. B. Poulsen, and S. Sedwards, "Importance sampling for stochastic timed automata," in *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016, Beijing, China, November 9-11, 2016, Proceedings*, ser. Lecture Notes in Computer Science, M. Fränzle, D. Kapur, and N. Zhan, Eds., vol. 9984, 2016, pp. 163–178. [Online]. Available: https://doi.org/10.1007/978-3-319-47677-3_11

[81] B. Jonsson and K. G. Larsen, "Specification and refinement of probabilistic processes," in *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*. IEEE Computer Society, 1991, pp. 266–277. [Online]. Available: https://doi.org/10.1109/LICS.1991.151651

[82] C. Jou and S. A. Smolka, "Equivalences, congruences, and complete axiomatizations for probabilistic processes," in *CONCUR '90, Theories of Concurrency: Unification and Extension, Amsterdam, The Netherlands, August 27-30, 1990, Proceedings*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. W. Klop, Eds., vol. 458. Springer, 1990, pp. 367–383. [Online]. Available: https://doi.org/10.1007/BFb0039071

[83] D. K. Kaynar, N. A. Lynch, R. Segala, and F. W. Vaandrager, *The Theory of Timed I/O Automata, Second Edition*, ser. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2010. [Online]. Available: https://doi.org/10.2200/S00310ED1V01Y201011DCT005

[84] S. Kiefer, "On computing the total variation distance of hidden Markov models," in *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, ser. LIPIcs, I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, Eds., vol. 107. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 130:1–130:13. [Online]. Available: https://doi.org/10.4230/LIPIcs.ICALP.2018.130

[85] D. Kozen, "Results on the propositional mu-calculus," *Theor. Comput. Sci.*, vol. 27, pp. 333–354, 1983. [Online]. Available: https://doi.org/10.1016/0304-3975(82)90125-6

[86] D. Kozen, R. Mardare, and P. Panangaden, "Strong completeness for Markovian logics," in *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and J. Sgall, Eds., vol. 8087. Springer, 2013, pp. 655–666. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40313-2_58

[87] J. Kretínský and A. Rotar, "The satisfiability problem for unbounded fragments of probabilistic CTL," in *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, ser. LIPIcs, S. Schewe and L. Zhang, Eds., vol. 118. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 32:1–32:16. [Online]. Available: https://doi.org/10.4230/LIPIcs.CONCUR.2018.32

[88] M. Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Verifying quantitative properties of continuous probabilistic timed automata," in *CONCUR 2000 - Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22-25, 2000, Proceedings*, ser. Lecture Notes in Computer Science, C. Palamidessi, Ed., vol. 1877. Springer, 2000, pp. 123–137. [Online]. Available: https://doi.org/10.1007/3-540-44618-4_11

[89] ——, "Automatic verification of real-time systems with discrete probability distributions," *Theor. Comput. Sci.*, vol. 282, no. 1, pp. 101–150, 2002. [Online]. Available: https://doi.org/10.1016/S0304-3975(01)00046-9

[90] F. Laroussinie and P. Schnoebelen, "The state explosion problem from trace to bisimulation equivalence," in *Foundations of Software Science and Computation Structures, Third International Conference, FOSSACS 2000, Held as Part of the Joint European Conferences on Theory and Practice of Software,ETAPS 2000, Berlin, Germany, March 25 - April 2, 2000, Proceedings*, ser. Lecture Notes in Computer Science, J. Tiuryn, Ed., vol. 1784. Springer, 2000, pp. 192–207. [Online]. Available: https://doi.org/10.1007/3-540-46432-8_13

[91] K. G. Larsen, U. Fahrenberg, and C. R. Thrane, "Metrics for weighted transition systems: Axiomatization and complexity," *Theor. Comput. Sci.*, vol. 412, no. 28, pp. 3358–3369, 2011. [Online]. Available: https://doi.org/10.1016/j.tcs.2011.04.003

[92] K. G. Larsen and R. Mardare, "Complete proof systems for weighted modal logic," *Theor. Comput. Sci.*, vol. 546, pp. 164–175, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2014.03.007

[93] K. G. Larsen, R. Mardare, and B. Xue, "Alternation-free weighted mu-calculus: Decidability and completeness," *Electr. Notes Theor. Comput. Sci.*, vol. 319, pp. 289–313, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2015.12.018

[94] ——, "Concurrent weighted logic," *J. Log. Algebr. Meth. Program.*, vol. 84, no. 6, pp. 884–897, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jlamp.2015.07.002

[95] ——, "On decidability of recursive weighted logics," *Soft Comput.*, vol. 22, no. 4, pp. 1085–1102, 2018. [Online]. Available: https://doi.org/10.1007/s00500-016-2193-z

[96] ——, "Decidability and expressiveness of recursive weighted logic," in *Perspectives of System Informatics - 9th International Ershov Informatics Conference, PSI 2014, St. Petersburg, Russia, June 24-27, 2014. Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Voronkov and I. Virbitskaite, Eds., vol. 8974. Springer, 2014, pp. 216–231. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-46823-4_18

[97] ——, "A decidable recursive logic for weighted transition systems," in *Theoretical Aspects of Computing - ICTAC 2014 - 11th International Colloquium, Bucharest, Romania, September 17-19, 2014. Proceedings*, ser. Lecture Notes in Computer Science, G. Ciobanu and D. Méry, Eds., vol. 8687. Springer, 2014, pp. 460–476. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10882-7_27

[98] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Inf. Comput.*, vol. 94, no. 1, pp. 1–28, 1991. [Online]. Available: https://doi.org/10.1016/0890-5401(91)90030-6

[99] K. G. Larsen and W. Yi, "Time-abstracted bisimulation: Implicit specifications and decidability," *Inf. Comput.*, vol. 134, no. 2, pp. 75–101, 1997. [Online]. Available: https://doi.org/10.1006/inco.1997.2623

[100] N. G. Leveson and C. S. Turner, "Investigation of the therac-25 accidents," *IEEE Computer*, vol. 26, no. 7, pp. 18–41, 1993. [Online]. Available: https://doi.org/10.1109/MC.1993.274940

[101] G. G. I. López, H. Hermanns, and J. Katoen, "Beyond memoryless distributions: Model checking semi-Markov chains," in *Process Algebra and Probabilistic Methods, Performance Modeling and Verification: Joint International Workshop, PAPM-PROBMIV 2001, Aachen, Germany, September 12-14, 2001, Proceedings*, ser. Lecture Notes in Computer Science, L. de Alfaro and S. Gilmore, Eds., vol. 2165. Springer, 2001, pp. 57–70. [Online]. Available: https://doi.org/10.1007/3-540-44804-7_4

[102] G. Lüttgen and W. Vogler, "A faster-than relation for asynchronous processes," in *CONCUR 2001 - Concurrency Theory, 12th International Conference, Aalborg, Denmark, August 20-25, 2001, Proceedings*, ser. Lecture Notes in Computer Science, K. G. Larsen and M. Nielsen, Eds., vol. 2154. Springer, 2001, pp. 262–276. [Online]. Available: https://doi.org/10.1007/3-540-44685-0_18

[103] ——, "Bisimulation on speed: A unified approach," *Theor. Comput. Sci.*, vol. 360, no. 1-3, pp. 209–227, 2006. [Online]. Available: https://doi.org/10.1016/j.tcs.2006.03.004

[104] R. Mardare, L. Cardelli, and K. G. Larsen, "Continuous Markovian logics - axiomatization and quantified metatheory," *Logical Methods in Computer Science*, vol. 8, no. 4, 2012. [Online]. Available: https://doi.org/10.2168/LMCS-8(4:19)2012

[105] I. Meinecke, "Weighted logics for traces," in *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, ser. Lecture Notes in Computer Science, D. Grigoriev, J. Harrison, and E. A. Hirsch, Eds., vol. 3967. Springer, 2006, pp. 235–246. [Online]. Available: http://dx.doi.org/10.1007/11753728_25

[106] H. Michalewski and M. Mio, "Baire category quantifier in monadic second order logic," in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, ser. Lecture Notes in Computer Science, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, Eds., vol. 9135. Springer, 2015, pp. 362–374. [Online]. Available: https://doi.org/10.1007/978-3-662-47666-6_29

[107] R. Milner, *Communication and concurrency*, ser. PHI Series in computer science. Prentice Hall, 1989.

[108] F. Moller and C. M. N. Tofts, "Relating processes with respect to speed," in *CONCUR '91, 2nd International Conference on Concurrency Theory, Amsterdam, The Netherlands, August 26-29, 1991, Proceedings*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. F. Groote, Eds., vol. 527. Springer, 1991, pp. 424–438. [Online]. Available: https://doi.org/10.1007/3-540-54430-5_104

[109] M. R. Neuhäußer and J. Katoen, "Bisimulation and logical preservation for continuous-time Markov decision processes," in *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires and V. T. Vasconcelos, Eds., vol. 4703. Springer, 2007, pp. 412–427. [Online]. Available: https://doi.org/10.1007/978-3-540-74407-8

[110] J. Ouaknine and J. Worrell, "On the language inclusion problem for timed automata: Closing a decidability gap," in *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*. IEEE Computer Society, 2004, pp. 54–63. [Online]. Available: https://doi.org/10.1109/LICS.2004.1319600

[111] D. M. R. Park, "Concurrency and automata on infinite sequences," in *Theoretical Computer Science, 5th GI-Conference, Karlsruhe, Germany, March 23-25, 1981, Proceedings*, ser. Lecture Notes in Computer Science, P. Deussen, Ed., vol. 104. Springer, 1981, pp. 167–183. [Online]. Available: https://doi.org/10.1007/BFb0017309

[112] M. Paterson, Ed., *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, July 16-20, 1990, Proceedings*, ser. Lecture Notes in Computer Science, vol. 443. Springer, 1990. [Online]. Available: https://doi.org/10.1007/BFb0032016

[113] M. R. Pedersen, G. Bacci, and K. G. Larsen, "A faster-than relation for semi-Markov decision processes," *CoRR*, vol. abs/1810.11243, 2018. [Online]. Available: https://arxiv.org/abs/1810.11243

[114] M. R. Pedersen, G. Bacci, K. G. Larsen, and R. Mardare, "A hemimetric extension of simulation for semi-Markov decision processes," in *Quantitative Evaluation of Systems - 15th International Conference, QEST 2018, Beijing, China, September 4-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science, A. McIver and A. Horvath, Eds., vol. 11024. Springer, 2018, pp. 339–355. [Online]. Available: https://doi.org/10.1007/978-3-319-99154-2_21

[115] M. R. Pedersen, N. Fijalkow, G. Bacci, K. G. Larsen, and R. Mardare, "Timed comparisons of semi-Markov processes," in *Language and Automata Theory and Applications - 12th International Conference, LATA 2018, Ramat Gan, Israel, April 9-11, 2018, Proceedings*, ser. Lecture Notes in Computer Science, S. T. Klein, C. Martín-Vide, and D. Shapira, Eds., vol. 10792. Springer, 2018, pp. 271–283. [Online]. Available: https://doi.org/10.1007/978-3-319-77313-1_21

[116] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1*

November 1977. IEEE Computer Society, 1977, pp. 46–57. [Online]. Available: https://doi.org/10.1109/SFCS.1977.32

[117] J. Queille and J. Sifakis, "Specification and verification of concurrent systems in CESAR," in *International Symposium on Programming, 5th Colloquium, Torino, Italy, April 6-8, 1982, Proceedings*, ser. Lecture Notes in Computer Science, M. Dezani-Ciancaglini and U. Montanari, Eds., vol. 137. Springer, 1982, pp. 337–351. [Online]. Available: https://doi.org/10.1007/3-540-11494-7_22

[118] M. O. Rabin, "Probabilistic automata," *Information and Control*, vol. 6, no. 3, pp. 230–245, 1963. [Online]. Available: https://doi.org/10.1016/S0019-9958(63)90290-0

[119] J. J. M. M. Rutten, "Universal coalgebra: a theory of systems," *Theor. Comput. Sci.*, vol. 249, no. 1, pp. 3–80, 2000. [Online]. Available: https://doi.org/10.1016/S0304-3975(00)00056-6

[120] I. Satoh and M. Tokoro, "A formalism for remotely interacting processes," in *Theory and Practice of Parallel Programming, International Workshop TPPP'94, Sendai, Japan, November 7-9, 1994, Proceedings*, ser. Lecture Notes in Computer Science, T. Ito and A. Yonezawa, Eds., vol. 907. Springer, 1994, pp. 216–228. [Online]. Available: https://doi.org/10.1007/BFb0026571

[121] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., 1986.

[122] M. P. Schützenberger, "On the definition of a family of automata," *Information and Control*, vol. 4, no. 2-3, pp. 245–270, 1961. [Online]. Available: https://doi.org/10.1016/S0019-9958(61)80020-X

[123] R. Segala and N. A. Lynch, "Probabilistic simulations for probabilistic processes," *Nord. J. Comput.*, vol. 2, no. 2, pp. 250–273, 1995.

[124] A. P. Sistla and E. M. Clarke, "The complexity of propositional linear temporal logics," *J. ACM*, vol. 32, no. 3, pp. 733–749, 1985. [Online]. Available: http://doi.acm.org/10.1145/3828.3837

[125] J. Sproston and A. Troina, "Simulation and bisimulation for probabilistic timed automata," in *Formal Modeling and Analysis of Timed Systems - 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and T. A. Henzinger, Eds., vol. 6246. Springer, 2010, pp. 213–227. [Online]. Available: https://doi.org/10.1007/978-3-642-15297-9_17

[126] S. Tasiran, R. Alur, R. P. Kurshan, and R. K. Brayton, "Verifying abstractions of timed systems," in *CONCUR '96, Concurrency Theory, 7th International Conference, Pisa, Italy, August 26-29, 1996, Proceedings*, ser. Lecture Notes in Computer Science, U. Montanari and V. Sassone, Eds., vol. 1119. Springer, 1996, pp. 546–562. [Online]. Available: https://doi.org/10.1007/3-540-61604-7_75

[127] J. van Benthem, "Modal correspondence theory," Ph.D. dissertation, Mathematish Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[128] F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell, "Domain theory, testing and simulation for labelled Markov processes," *Theor. Comput. Sci.*, vol. 333, no. 1-2, pp. 171–197, 2005. [Online]. Available: https://doi.org/10.1016/j.tcs.2004.10.021

[129] F. van Breugel and J. Worrell, "A behavioural pseudometric for probabilistic transition systems," *Theor. Comput. Sci.*, vol. 331, no. 1, pp. 115–142, 2005. [Online]. Available: https://doi.org/10.1016/j.tcs.2004.09.035

[130] ——, "Approximating and computing behavioural distances in probabilistic transition systems," *Theor. Comput. Sci.*, vol. 360, no. 1-3, pp. 373–385, 2006. [Online]. Available: https://doi.org/10.1016/j.tcs.2006.05.021

[131] R. J. van Glabbeek, S. A. Smolka, and B. Steffen, "Reactive, generative and stratified models of probabilistic processes," *Inf. Comput.*, vol. 121, no. 1, pp. 59–80, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1123

[132] M. Y. Vardi, "Automatic verification of probabilistic concurrent finite-state programs," in *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*. IEEE Computer Society, 1985, pp. 327–338. [Online]. Available: https://doi.org/10.1109/SFCS.1985.12

[133] M. Y. Vardi and P. Wolper, "Reasoning about infinite computations," *Inf. Comput.*, vol. 115, no. 1, pp. 1–37, 1994. [Online]. Available: https://doi.org/10.1006/inco.1994.1092

[134] W. Vogler, "Timed testing of concurrent systems," *Inf. Comput.*, vol. 121, no. 2, pp. 149–171, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1130

[135] ——, "Faster asynchronous systems," *Inf. Comput.*, vol. 184, no. 2, pp. 311–342, 2003. [Online]. Available: https://doi.org/10.1016/S0890-5401(03)00065-8

[136] V. Wolf, C. Baier, and M. E. Majster-Cederbaum, "Trace machines for observing continuous-time Markov chains," *Electr. Notes Theor. Comput. Sci.*, vol. 153, no. 2, pp. 259–277, 2006. [Online]. Available: https://doi.org/10.1016/j.entcs.2005.10.042

[137] L. Zhang, "Decision algorithms for probabilistic simulations," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2009. [Online]. Available: http://scidok.sulb.uni-saarland.de/volltexte/2009/2424/

Chapter 2

Preliminaries

In this chapter we introduce some of the key concepts and standard results that we will use throughout the paper. The material in this chapter is not novel, and can be found in any standard textbook on each of the subjects discussed.

We will use $\mathbb{N}$, $\mathbb{Q}$, and $\mathbb{R}$ to denote the natural, rational, and real numbers, respectively. Furthermore, we will use $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ to denote the non-negative rational and real numbers, respectively, and $\mathbb{R}_{>0}$ denotes the strictly positive real numbers. For expressions involving ∞, we will adopt the convention that

$$\infty + x = x + \infty = \infty$$

whenever $x \in \mathbb{R}$ and

$$\infty \cdot x = x \cdot \infty = \infty$$

whenever $x \in \mathbb{R}_{>0}$.

2.1 Set Theory

We assume the reader is familiar with the basic concepts of set theory. Given two sets A and B, $A \cup B$, $A \cap B$, $A \times B$ is the union, intersection, and Cartesian product, respectively, of A and B. We denote by 2^A the power set of A and by A^c the complement of A. For a function $f : A \to B$, the preimage of a set $Y \subseteq B$ under f is given by

$$f^{-1}(Y) = \{x \in A \mid f(x) \in Y\}.$$

A *relation* on a set A is simply a subset $R \subseteq A \times A$. We will sometimes write aRb to mean $(a, b) \in R$. An *equivalence relation* on a set A is a relation $R \subseteq A \times A$ that has the following properties.

Reflexivity: $(a, a) \in R$ for any $a \in A$.

Symmetry: If $(a, b) \in R$, then $(b, a) \in R$.

Transitivity: If $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$.

An equivalence relation partitions a set into *equivalence classes* such that every element of the set is in exactly one equivalence class. A *preorder* is a relation which satisfies reflexivity and transitivity.

A set which can be put into bijection with a subset of the natural numbers is said to be *countable*, otherwise it is said to be *uncountable*.

2.2 Boolean Algebra

Boolean algebra is a formalisation and generalisation of the rules of logic as initially introduced by George Boole. For a comprehensive introduction to Boolean algebras, see the excellent textbook by Givant and Halmos [6].

Definition 2.2.1. A *Boolean algebra* is a set A together with two binary operations $\wedge$ and $\vee$, a unary operation $\neg$, and two distinguished elements $\top$ and $\bot$. These must satisfy the following conditions, for any elements $p, q, r \in A$.

$$p \wedge \top = p \qquad p \vee \bot = p \qquad \text{(identity laws)}$$

$$p \wedge \neg p = \bot \qquad p \vee \neg p = \top \qquad \text{(complement laws)}$$

$$p \wedge q = q \wedge p \qquad p \vee q = q \vee p \qquad \text{(commutative laws)}$$

$$\begin{aligned} p \wedge (q \vee r) &= (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) &= (p \vee q) \wedge (p \vee r) \end{aligned} \qquad \text{(associative laws)}$$

▲

The two operations $\wedge$ and $\vee$ are known as *meet* and *join*, respectively, whereas the operation $\neg$ is known as *complement*. The elements $\top$ and $\bot$ are known as the *top* and *bottom* elements, respectively.

Example 2.2.2. The two-element Boolean algebra has only the elements $\top$ and $\bot$. In this algebra, we interpret $\top$ as true and $\bot$ as false. We then recover the usual logical connectives, as $\wedge$ becomes conjunction, $\vee$ becomes disjunction, and $\neg$ becomes negation. ⧫

Example 2.2.3. The power set 2^X of a set X is a Boolean algebra where meet is intersection, join is union, and complement is set complement. ⧫

From the definition of a Boolean algebra follows naturally a notion of *order* between the elements of a Boolean algebra.

Definition 2.2.4. We will write $p \to q$ and say that p is *below* q if $p \wedge q = p$. ▲

This order also explains the names top and bottom for $\top$ and $\bot$, since we now have $\bot \to p \to \top$ for any $p \in A$, meaning that $\bot$ is below every element and every element is below $\top$.

A map from one Boolean algebra to another that preserves the structure of the Boolean algebra is called a homomorphism.

Definition 2.2.5. Let A and B be two Boolean algebras. A *homomorphism* from A to B is a map $f : A \to B$ such that

$$\begin{aligned} f(p \wedge q) &= f(p) \wedge f(q) \\ f(p \vee q) &= f(p) \vee f(q) \\ f(\neg p) &= \neg f(p) \end{aligned}$$ ▲

It is simple to show that if f is a homomorphism, then we also get

$$f(\top) = \top \quad \text{and} \quad f(\bot) = \bot.$$

An important example of homomorphism comes from the quotient constructed from a congruence.

Definition 2.2.6. A *congruence* on a Boolean algebra A is an equivalence relation $\mathcal{R}$ such that whenever $p\mathcal{R}r$ and $q\mathcal{R}s$ we also have

$$\begin{aligned} (p \wedge q)\mathcal{R}(r \wedge s) \\ (p \vee q)\mathcal{R}(r \vee s) \\ (\neg p)\mathcal{R}(\neg r) \end{aligned}$$ ▲

Given a congruence $\mathcal{R}$ on a Boolean algebra A, we can now construct the *quotient* of A under $\mathcal{R}$, which is denoted by $A/\mathcal{R}$. The quotient of A under $\mathcal{R}$ consists of all equivalence classes of $\mathcal{R}$, and is in fact a Boolean algebra by defining meet, join, and complement as

$$\begin{aligned} [p] \wedge [q] &= [p \wedge q] \\ [p] \vee [q] &= [p \vee q] \\ \neg[p] &= [\neg p], \end{aligned}$$

where $[p]$ denotes the equivalence class of p.

Proposition 2.2.7 ([6, Chapter 17])**.** *Let A be a Boolean algebra and $\mathcal{R}$ a congruence on A. Define the function $f : A \to A/\mathcal{R}$ by $f(p) = [p]$. Then f is a homomorphism, known as the* projection *from A unto $A/\mathcal{R}$.*

Example 2.2.8. Consider a Boolean algebra A. Define $p \leftrightarrow q$ if and only if $p \to q$ and $q \to p$. For example, it is easy to see that $(p \wedge \top) \leftrightarrow p$. Then the relation

$$\mathcal{R}_{\leftrightarrow} = \{(p, q) \in A \times A \mid p \leftrightarrow q\},$$

is an equivalence relation and can be shown to be a congruence on A. The quotient of A under $\mathcal{R}_{\leftrightarrow}$ is known as the *Lindenbaum* algebra. ⧫

An important concept for Boolean algebras is that of a filter.

Definition 2.2.9. Given a Boolean algebra A, a *filter* is a subset $F \subseteq A$ such that

1. $\top \in F$,
2. if $p \in F$ and $q \in F$, then $p \wedge q \in F$, and
3. If $p \in F$ and $p \to q$, then $q \in F$. ▲

In other words, a filter is a subset which contains $\top$, is closed under meet, and is upward-closed.

Definition 2.2.10. Given a Boolean algebra A, U is an *ultrafilter* if

- U is a filter,
- $U \neq A$, and
- if F is another filter such that $U \subseteq F$, then either $F = U$ or $F = A$. ▲

An ultrafilter is therefore a filter which is maximal, in the sense that we can not add anything to the filter while having it remain a filter. Ultrafilters have the following nice property.

Lemma 2.2.11 ([6, Chapter 20, Lemma 1])**.** *Let A be a Boolean algebra. U is an ultrafilter if and only if for any $p \in A$ we have either $p \in U$ or $\neg p \in U$, but not both.*

2.3 Metric Spaces

The theory of metric spaces is concerned with notions of distance. The most basic distance is that of Euclidean distance, which is simply the length of the straight line between two points in a Euclidean space. However, this notion quickly becomes too simplistic. For example, the distance one has to travel to go from Denmark to England depends on whether one is going by airplane, by ferry, by train, or something else. Furthermore, the Euclidean distance is symmetric: If the distance from A to B is x, then the distance from B to A is also x. However, some natural notions of distance are not symmetric. For

example, the distance when traveling by car in a city from point A to point B may not be the same as the distance from B to A, due to the existence of one-way streets.

Definition 2.3.1. Let X be a set, and $d : X \times X \to [0, \infty]$ a function. Consider the following conditions on d.

(D1) $d(x, y) = 0$ implies $x = y$.

(D2) $x = y$ implies $d(x, y) = 0$.

(D3) $d(x, y) = d(y, x)$.

(D4) $d(x, z) \leq d(x, y) + d(y, z)$.

The function d is

- a *metric* if it satisfies (D1)-(D4),
- a *pseudometric* if it satisfies (D2)-(D4), and
- a *hemimetric* if it satisfies (D2) and (D4). ▲

Condition (D3) is known as symmetry and condition (D4) is known as the triangle inequality. We will use the term *distance* to mean any of the above three.

Example 2.3.2. For any set X, define the function

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

This is known as the *discrete* metric. ♦

Example 2.3.3. Consider the real numbers $\mathbb{R}$ and define the function

$$d(x, y) = |y - x| = \sqrt{(y - x)^2}.$$

This is known as the *Euclidean distance*, and simply measures the length of the straight line between the two points x and y. This can easily be generalised to $\mathbb{R}^n$ for any n. ♦

A metric space is then a set X together with a metric on X, and similarly for a pseudometric and hemimetric space. Metric spaces are the most common and well-studied of the three types of spaces in the literature. However, for us, the pseudometric and hemimetric spaces are more natural. This is because we want systems that have the same behaviour to be at distance 0 from each other, but having the same behaviour does not necessarily mean that the systems are equal. Hence condition (D1) becomes unnatural.

Given a distance d on X, we can define the closed and open sets of X as follows. For a point $x \in X$ and a radius $r > 0$, the open ball of radius r centered in x is

$$\mathcal{B}_r(x) = \{y \in X \mid d(x,y) < r\}. \tag{2.1}$$

Definition 2.3.4. Given a space X with a distance d, a subset $U \subseteq X$ is said to be *open* if for any $x \in U$ there exists $r > 0$ such that $\mathcal{B}_r(x) \subseteq U$. A subset $V \subseteq X$ is said to be *closed* if its complement V^c is open. ▲

In other words, a set is open if for any point in the set, we can find an open ball around that point such that the open ball is contained in the set. Note that a set can be both open and closed at the same time, such a set is called *clopen*.

The open sets (or equivalently, the closed sets) of a space X with a distance d form a topology on X.

Definition 2.3.5. Given a set X, a *topology* on X is a collection of subsets $\tau \subseteq 2^X$ such that

- $\emptyset \in \tau$ and $X \in \tau$,
- any union of elements of τ is again in τ, and
- any finite intersection of elements of τ is again in τ. ▲

A topological space is then a set X together with a topology τ on X. Topological spaces are more general than metric, pseudometric, or hemimetric spaces, since the distance of a metric, pseudometric, or hemimetric space induces a topology, and thus these are also topological spaces, whereas a topological space need not have a distance at all.

In Equation (2.1), we used $d(x,y)$, the distance from x to y, for the definition of an open ball. For metric and pseudometric spaces, it would not have made a difference if we instead had written $d(y,x)$, the distance from y to x, since in these spaces, the distance is symmetric. However, for hemimetric spaces, we will distinguish between the *left-centered open balls* $\mathcal{B}^L_r(x)$ and the *right-centered open balls* $\mathcal{B}^R_r(x)$, defined as

$$\mathcal{B}^L_r(x) = \{y \in X \mid d(x,y) < r\} \quad \text{and} \quad \mathcal{B}^R_r(x) = \{y \in X \mid d(y,x) < r\}.$$

The left-centered and the right-centered open balls both give rise to a topology, but these two topologies will in general be quite different.

The intuition for open sets is that they do not necessarily contain their border, whereas closed sets must contain their border, so that whenever the points in the set get infinitely close to some other point, then that other point must also be in the set. To make this intuition precise, we introduce the notion of convergence.

Definition 2.3.6. Let X be a space with a distance d. We say that a sequence of points $(x_n)_{n\in\mathbb{N}}$ *converges* to $x \in X$ if for every $\varepsilon > 0$ we can find $N \in \mathbb{N}$ such that for every $n \geq N$ we have that $d(x, x_n) < \varepsilon$. ▲

If a sequence $(x_n)_{n\in\mathbb{N}}$ converges to x, we will also say that x is a *limit* of $(x_n)_{n\in\mathbb{N}}$. In metric spaces limits are unique, so that we may speak of *the* limit of a sequence. However, this is not the case for pseudometric and hemimetric spaces. We can now define what it means to be sequentially closed.

Definition 2.3.7. A set $U \subseteq X$ is *sequentially closed* if whenever a sequence $(x_n)_{n\in\mathbb{N}}$, where $x_n \in U$ for all $n \in \mathbb{N}$, converges to some x, then also $x \in U$. ▲

A set is therefore sequentially closed if the limit of any sequence is again in that set. The two notions of closed set and sequentially closed set coincide.

Lemma 2.3.8 ([7, Exercise 4.7.14 and Lemma 6.3.6])**.** *For metric, pseudometric, and hemimetric spaces, a set is closed if and only if it is sequentially closed.*

2.4 Measure Theory

The aim of measure theory is to generalise the notion of size by "measuring" the size of sets. This is usually simple for finite and countable sets, but becomes very subtle for uncountable sets such as the real numbers. We will only concern ourselves here with those measures that assign a probability to sets, the so-called probability measures. The starting point of measure theory is the notion of a σ-algebra.

Definition 2.4.1. Let X be a set. A *σ-algebra* on X is a non-empty collection of subsets $\Sigma \subseteq 2^X$ such that

(A1) $X \in \Sigma$,

(A2) $A \in \Sigma$ implies $A^c \in \Sigma$, and

(A3) $A_1, A_2, A_3, \cdots \in \Sigma$ implies $\bigcup_{n=1}^{\infty} A_n \in \Sigma$.

A *measurable space* is a set X together with a σ-algebra on X. ▲

Condition (A2) is known as closure under complement, and condition (A3) is known as closure under countable union. Because we have closure under complement, condition (A1) could be replaced by $\emptyset \in \Sigma$. Furthermore, conditions (A2) and (A3) together also imply closure under countable intersection. An element of a σ-algebra will be called a *measurable set*.

Example 2.4.2. For any set X, $\{\emptyset, X\}$ is a σ-algebra on X. This is known as the *trivial* or *indiscrete* σ-algebra on X. ♦

Example 2.4.3. For any set X, the power set 2^X is a σ-algebra on X, known as the *discrete* σ-algebra on X. ⧫

Example 2.4.4. If we start with a topological space X with topology τ, then the *Borel σ-algebra* is the smallest σ-algebra containing all the elements of τ. The elements of the Borel σ-algebra are called *Borel sets*. ⧫

When speaking about the real numbers, we will always assume that they are equipped with the Borel σ-algebra, which we denote by $\mathbb{B}$.

The structure-preserving maps between measurable spaces are the measurable functions.

Definition 2.4.5. Given two measurable spaces X and Y with σ-algebra Σ_X and Σ_Y, respectively, a function $f : X \to Y$ is *measurable* if $f^{-1}(E) \in \Sigma_X$ for any $E \in \Sigma_Y$. ▲

We will now introduce the central concept of a measure. A measure assigns a numerical value to each measurable set, which we may interpret as the size of that set.

Definition 2.4.6. Given a measurable space X with σ-algebra Σ, a *measure* is a function $\mu : \Sigma \to \mathbb{R}_{\geq 0}$ such that

(M1) $\mu(\emptyset) = 0$ and

(M2) for any countable collection $A_1, A_2, A_3, \cdots \in \Sigma$ of pairwise disjoint sets it holds that

$$\mu\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \mu(A_i).$$

A measure $\mu : \Sigma \to [0, 1]$ is called a *subprobability measure* if $\mu(X) \leq 1$ and a *probability measure* if $\mu(X) = 1$.

Given a set X, we will use the notation

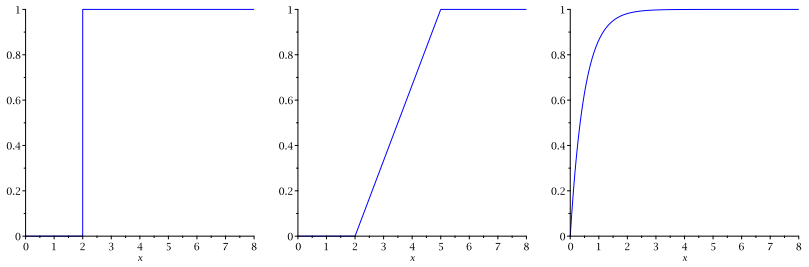
- $\mathcal{D}(X)$ to denote the set of all subprobability measures on X, and
- $\mathcal{D}_{=1}(X)$ to denote the set of all probability measures on X. ▲

Condition (M2) is known as countable additivity. A probability measure is thus a function that assigns a probability to the measurable sets, with the condition that the probability of *something* happening must be 1. A measure $\mu \in \mathcal{D}(X)$ will be said to be *finitely supported* if its support

$$\text{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$$

is finite.

An important example of measure is the product measure.



(a) Dirac **(b)** Uniform **(c)** Exponential

Figure 2.4.1: Plots of the CDFs of a Dirac distribution at 2, a uniform distribution with parameters $a = 2$ and $b = 5$, and an exponential distribution with rate 2.

Proposition 2.4.7 (Product measure [4, Theorem 18.2])**.** *Let X and Y be two measurable spaces with σ-algebra Σ_X and Σ_Y, respectively. We will then denote by $X \times Y$ the* product space*, which is a measurable space with σ-algebra $\Sigma_X \otimes \Sigma_Y$, defined as the smallest σ-algebra containing the sets $E \times F$ for $E \in \Sigma_X$ and $F \in \Sigma_Y$.*

Given two measures $\mu : \Sigma_X \to \mathbb{R}_{\geq 0}$ and $\nu : \Sigma_Y \to \mathbb{R}_{\geq 0}$, the product measure *$\mu \times \nu : \Sigma_X \otimes \Sigma_Y \to \mathbb{R}_{\geq 0}$, is the unique measure such that*

$$(\mu \times \nu)(E \times F) = \mu(E) \cdot \nu(F)$$

for all $(E, F) \in \Sigma_X \times \Sigma_Y$.

One of the important concepts derived from a probability measure is the cumulative distribution function.

Definition 2.4.8. Given a probability measure $\mu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, the *cumulative distribution function (CDF)*, or simply *distribution function*, of μ will be denoted by F_μ and is given by $F_\mu(t) = \mu([0, t])$. ▲

Consider now (sub)probability measures on the non-negative real numbers, i.e. $\mu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, where we interpret $\mathbb{R}_{\geq 0}$ as time. If we consider $\mu(X)$ for some measurable X to be the probability that an event, such as a system taking an action, has happened within the time interval given by X. Then $F_\mu(t)$ gives the probability that an event has occurred before the time point t. CDFs have the following nice properties.

Monotonicity: If $x \leq y$, then $F_\mu(x) \leq F_\mu(y)$.

Right-continuity: Let $x \in \mathbb{R}_{\geq 0}$. For every $\varepsilon > 0$ there exists a $\delta > 0$ such that if $x < y < x + \delta$, then $F_\mu(y) - F_\mu(x) < \varepsilon$.

Example 2.4.9. The *Dirac measure at x*, denoted by δ_x, is given by

$$\delta_x(E) = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{if } x \notin E \end{cases}$$

for any measurable E. We will denote the CDF of δ_x by $Dirac[x]$, which has the property that

$$Dirac[x]\,(t) = \begin{cases} 1 & \text{if } t \geq x \\ 0 & \text{otherwise.} \end{cases}$$

The CDF of the Dirac measure at 2 is plotted in Figure 2.4.1a. ⧫

Example 2.4.10. A *uniform* distribution is given by two parameters $a, b \in \mathbb{R}_{\geq 0}$ such that $a < b$. We will denote its CDF by $Unif[a, b]$, which is defined by

$$Unif[a, b]\,(t) = \begin{cases} 1 & \text{if } t < a \\ \frac{t-a}{b-a} & \text{if } a \leq t < b \\ 0 & \text{if } x \geq b. \end{cases}$$

The CDF of a uniform distribution with parameters $a = 2$ and $b = 5$ is plotted in Figure 2.4.1b. ⧫

Example 2.4.11. An *exponential* distribution is given by a parameter $\theta > 0$, often called the *rate*. Its CDF will be denoted by $Exp[\theta]$, and is defined by

$$Exp[\theta]\,(t) = 1 - e^{-\theta t}.$$

The CDF of an exponential distribution with rate 2 is plotted in Figure 2.4.1c. ⧫

Since it is often difficult to define a measure directly on a σ-algebra, some of the most useful results in measure theory are the extension theorems that allow us to only define something simpler, after which the extension theorem guarantees us that this definition can be extended to a measure on the σ-algebra. The extension theorem that we are interested in requires the following definition.

Definition 2.4.12. Let A be a Boolean algebra of sets. A function $\mu_0 : A \to \mathbb{R}_{\geq 0}$ is called a *pre-measure* if

(P1) $\mu_0(\emptyset) = 0$ and

(P2) $\mu_0(\bigcup_{n=1}^{\infty} E_n) = \sum_{n=1}^{\infty} \mu_0(E_n)$ whenever $E_1, E_2, \cdots \in A$ are disjoint sets such that $\bigcup_{n=1}^{\infty} E_n \in A$. ▲

A pre-measure is therefore much like a measure, except that it is defined on a Boolean algebra rather than a σ-algebra. However, by the following theorem, any pre-measure can be uniquely extended to a measure on a σ-algebra.

Theorem 2.4.13 (Hahn-Kolmogorov Theorem [12, Theorem 1.7.8])**.** *Let A be a Boolean algebra of sets. Any pre-measure $\mu_0 : A \to \mathbb{R}_{\geq 0}$ can be uniquely extended to a measure $\mu : \Sigma \to \mathbb{R}_{\geq 0}$, where Σ is the smallest σ-algebra containing A, such that $\mu_0(E) = \mu(E)$ whenever $E \in A$.*

2.4.1 Integration

An important part of measure theory is the theory of integration. We will not concern ourselves here with the intricacies of defining integration through measure theory. Instead, we will state some properties of the Lebesgue integral that are useful when manipulating integrals. In the following, we therefore assume that all functions are integrable. We use the notation

$$\int_E f(x)\ \mu(\mathrm{d}x)$$

to denote that we are integrating the function f over the measurable set E with respect to the measure μ viewed as a function of x. When $E = [0, t]$, we instead write

$$\int_0^t f(x)\ \mu(\mathrm{d}x).$$

Lemma 2.4.14 (Linearity of integrals [4, Theorem 16.1(ii)])**.** *The integral is linear, meaning that*

$$\int_E af(x) + bg(x)\ \mu(\mathrm{d}x) = a\int_E f(x)\ \mu(\mathrm{d}x) + b\int_E g(x)\ \mu(\mathrm{d}x)$$

when $a, b \in \mathbb{R}_{\geq 0}$ and E is measurable.

Lemma 2.4.15 (Fubini's theorem [9, Theorem 3.16])**.** *Let X and Y be measurable spaces with σ-algebra Σ_X and Σ_Y, respectively, and let $\mu : \Sigma_X \to \mathbb{R}_{\geq 0}$ and $\nu : \Sigma_Y \to \mathbb{R}_{\geq 0}$ be measures. If $f : X \times Y \to [0, \infty]$ is a measurable function, then*

$$\begin{aligned}\int_{E\times F} f(x,y)\ (\mu\times\nu)(\mathrm{d}(x,y)) &= \int_E\int_F f(x,y)\ \nu(\mathrm{d}y)\mu(\mathrm{d}x)\\ &= \int_F\int_E f(x,y)\ \mu(\mathrm{d}x)\nu(\mathrm{d}y)\end{aligned}$$

for any $E \in \Sigma_X$ and $F \in \Sigma_Y$.

Lemma 2.4.16 (Change of variable [9, Proposition 3.8])**.** *Let X and Y be measurable spaces with σ-algebra Σ_X and Σ_Y, respectively. Furthermore, let $T : X \to Y$ be a measurable function, and define the measure ν by $\nu = \mu \circ T^{-1}$.*

If $f : Y \to \mathbb{R}_{\geq 0}$ is a measurable function, then

$$\int_{T^{-1}(E)} (f \circ T)(x)\ \mu(\mathrm{d}x) = \int_E f(y)\ \nu(\mathrm{d}y)$$

for any $E \in \Sigma_Y$.

Using integration, we can now define the important concept of convolution. Whereas the CDF $F_\mu(t)$ gives the probability that a single event has occurred before time t, we are often interested in the probability that multiple events have all occurred before time t. For this, we need the notion of convolution.

Definition 2.4.17. Given two measures $\mu, \nu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, the *convolution* of μ and ν is given by

$$(\mu * \nu)([0,t]) = \int_0^t \nu([0,t-x])\ \mu(\mathrm{d}x). \qquad \blacktriangle$$

$(\mu * \nu)([0,t])$ is then the probability that the event governed by μ and the event governed by ν have both occurred, in sequence, before time t. Convolution is both commutative and associative, meaning that

$$\mu * \nu = \nu * \mu \quad \text{and} \quad \mu * (\nu * \eta) = (\mu * \nu) * \eta.$$

Furthermore, the Dirac measure at 0 is the identity for convolution, so that

$$\mu * \delta_0 = \mu.$$

2.5 Complexity Theory

One of the most important concepts in computer science is that of an algorithm. Informally, an algorithm is a mechanical procedure for producing a set of outputs. Many equivalent ways of formalising the notion of algorithm have been proposed, most notably those of Turing machines, recursive functions, and the λ-calculus. We will take our underlying model of computation to be that of Turing machines. However, as is commonly done, we will describe algorithms informally as pseudocode, with the understanding that such pseudocode could, if needed, be translated into one of the abovementioned formalisms. The textbook by Sipser provides a gentle introduction to the theory of computability and complexity [11].

One of the most profound results of computability theory is the fact that, for some problems, there can not exist an algorithm to solve that problem.

Definition 2.5.1. A *decision problem* is the problem of deciding whether a given element is a member of some set. A decision problem is said to be *decidable* if there exists an algorithm that solves the decision problem. If no such algorithm exists, the decision problem is said to be *undecidable*. $\blacktriangle$

Sometimes we want our algorithm to not just answer yes or no, but to compute some value, for example the value of a function. For this, we have the notion of computability.

Definition 2.5.2. We will say that a function is *computable* if there exists an algorithm which computes the value of the function for any input in the domain of the function. $\blacktriangle$

For practical issues, we are not only interested in whether or not an algorithm exists, but also in how fast that algorithm runs: Does it finish in

seconds, or do we have to wait years for the output? In order to talk about the running time of algorithms, we need the following notation.

Definition 2.5.3. Given two functions $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we will say that g is an *asymptotic upper bound* for f and write $f(n) \in \mathcal{O}(g(n))$ if we can find $c, N \in \mathbb{N}$ such that for any $n \geq N$ we have $f(n) \leq cg(n)$. ▲

Here, the word "asymptotic" means "as we approach infinity", meaning that the function g will eventually become an upper bound for the function f, if we let n become large enough. $\mathcal{O}$ therefore suppresses smaller terms and constant factors.

Example 2.5.4. When f is a polynomial, we can simply pick out the largest term in the polynomial, which will then be an asymptotic upper bound for f. For example, if $f(n) = 7n^4 + 3n^2 + 5$, then $7n^4$ is the largest term, and we can drop the constant 7, so we obtain that $f(n) \in \mathcal{O}(n^4)$. ⧫

We can now use the concept of asymptotic upper bound to describe the complexity of algorithms.

Definition 2.5.5. We will say that an algorithm *uses time* $\mathcal{O}(g(n))$ if $f(n) \in \mathcal{O}(g(n))$, where f is a function that returns the number of steps that the algorithm goes through when given an input of size n.

Likewise, we will say that an algorithm *uses space* $\mathcal{O}(g(n))$ if $f(n) \in \mathcal{O}(g(n))$, where f is a function that returns the amount of space or memory that the algorithm uses when given an input of size n. ▲

We can now classify the complexity of different algorithms, leading to a veritable zoo of complexity classes [1]. We will describe here only some of the most important complexity classes. Let $g(n)$ be a polynomial function.

P (or PTIME) is the class of problems that can be solved by an algorithm that uses $\mathcal{O}(g(n))$ time.

NP is the class of problems such that whenever the answer is "yes", there exists a proof or witness of this fact, and furthermore, there exists an algorithm in **P** that can verify whether a given proof is correct.

coNP is the class of problems such that whenever the answer is "no", there exists a proof or witness of this fact, and furthermore, there exists an algorithm in **P** that can verify whether a given proof is correct.

PSPACE is the class of problems that can be solved by an algorithm that uses $\mathcal{O}(g(n))$ space.

EXPTIME is the class of problems that can be solved by an algorithm that uses $\mathcal{O}(2^{g(n)})$ time.

EXPSPACE is the class of problems that can be solved by an algorithm that uses $\mathcal{O}(2^{g(n)})$ space.

2.6 Models

In this thesis we will make use of two different kinds of models: Weighted transition systems and semi-Markov processes, both of which are standard in the literature. Weighted transition systems are similar to labelled transitions systems [3] except they have weights on transitions instead of labels. Semi-Markov processes extend continuous-time Markov processes by allowing the sojourn time to follow any distribution, not just exponential distributions [10].

2.6.1 Weighted Transition Systems

Weighted transition systems are systems in which each transition from a state to a new state is associated with some weight. This weight can be interpreted as the cost of taking that transition, the time spent taking that transition, the amount of resources spent taking that transition, etc. Assume that we have a countable set of *atomic propositions*, denoted by $\mathcal{AP}$.

Definition 2.6.1. A *weighted transition system (WTS)* is a tuple $M = (S, \to, \ell)$, where

1. S is a set of *states*,
2. $\to \subseteq S \times \mathbb{R}_{\geq 0} \times S$ is the *transition relation*, and
3. $\ell : S \to 2^{\mathcal{AP}}$ is the *labelling function*. ▲

We will write $s \xrightarrow{r} t$ to mean that $(s, r, t) \in \to$. A WTS is said to be *image-finite* if for any $s \in S$, there are only finitely many $t \in S$ such that $s \xrightarrow{r} t$ for some $r \in \mathbb{R}_{\geq 0}$.

A WTS operates by non-deterministically choosing, from a state s, a transition $s \xrightarrow{r} t$, after which the system ends up in state t, and from there can do further transitions. In a state s, $\ell(s)$ gives all the atomic propositions that are true in that state. One can also think of $\ell(s)$ as the labels that we put on that state.

Example 2.6.2. An example of a WTS was given in Example 1.3.1. In this example, we have three states, so $S = \{s_1, s_2, s_3\}$, where $\ell(s_1) = \{\texttt{waiting}\}$, $\ell(s_2) = \{\texttt{cleaning}\}$, and $\ell(s_3) = \{\texttt{charging}\}$. The transition relation is given by the arrows between the states, so that e.g. $s_1 \xrightarrow{0} s_2$ and $s_3 \xrightarrow{60} s_1$. ♦

The standard way of determining whether two WTSs behave the same is that of bisimulation [5].

Definition 2.6.3. Let $M = (S, \to, \ell)$ be a WTS. A *weighted bisimulation relation* is an equivalence relation $\mathcal{R} \subseteq S \times S$ such that $(s, t) \in \mathcal{R}$ implies

(Atomic harmony) $\ell(s) = \ell(t)$,

(Zig) if $s \xrightarrow{r} s'$, then there exists $t' \in S$ such that $t \xrightarrow{r} t'$ and $(s', t') \in \mathcal{R}$, and

(Zag) if $t \xrightarrow{r} t'$, then there exists $s' \in S$ such that $s \xrightarrow{r} s'$ and $(s', t') \in \mathcal{R}$. ▲

We will say that $s, t \in S$ are *weighted bisimilar* and write $s \sim_W t$ if there exists a weighted bisimulation relation $\mathcal{R}$ such that $(s, t) \in \mathcal{R}$. Weighted bisimilarity, denoted by $\sim_W$, is the largest weighted bisimulation.

Atomic harmony says that any states in the relation must have the same labels. The zig and zag conditions are responsible for ensuring that the transition behaviour of the two states is the same. Zig says that if s can do a transition, then t can do the same transition, and the states that we end up in after taking these transitions are also bisimilar. Zag is symmetric, saying that s can match transitions taken by t.

2.6.2 Semi-Markov Processes

A semi-Markov process is a system in which both the time that is spent in each state and the next state reached after taking a transition is determined probabilistically. We will take here the view of semi-Markov processes as graphs or transition systems, rather than as a sequence of random variables. This also means that we will implicitly assume that all semi-Markov processes are time-homogeneous. We assume a countable set $\texttt{In}$ of input actions and a countable set $\texttt{Out}$ of output actions.

Definition 2.6.4. A *semi-Markov process (SMP)* is a tuple $\mathcal{M} = (S, \tau, \rho, \ell)$, where

1. S is a countable set of *states*,
2. $\tau : S \times \texttt{In} \to \mathcal{D}(S \times \texttt{Out})$ is the *transition function*,
3. $\rho : S \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is the *time-residence function*, and
4. $\ell : S \to 2^{\mathcal{AP}}$ is the *labelling* function. ▲

The operational behaviour of a SMP is as follows. Starting in a state s, the SMP receives some input a from the environment. It then probabilistically goes to a new state s' while outputting some b after waiting some time t, the probability of which is given by $\tau(s, a)(s', b) \cdot \rho(s)([0, t])$. The labelling function tells us which labels each state has. For a state $s \in S$, we will write F_s for the CDF $\rho(s)$, i.e. $F_s(t) = \rho(s)([0, t])$.

We will say that an SMP $\mathcal{M} = (S, \tau, \rho, \ell)$ is *finite* if S is a finite set. If $\mathcal{M}$ is finite, we will denote by $|\mathcal{M}|$ the size of the state space of $\mathcal{M}$.

Example 2.6.5. An example of a SMP was given in Example 1.3.2. Here we have four states $S = \{s_1, s_2, s_3, s_4\}$, where s_1 is the top left state, s_2 the top right state, s_3 the bottom left state, and s_4 the bottom left state. ρ is then given by

$$\rho(s_1) = \rho(s_3) = Exp[0.1] \quad \text{and} \quad \rho(s_2) = \rho(s_4) = Exp[2]\,,$$

and ℓ is given by

$$\ell(s_1) = \ell(s_2) = \{g_1, r_2\} \quad \text{and} \quad \ell(s_3) = \ell(s_4) = \{r_1, g_2\}.$$

The transition function τ is given by the arrows in Figure 1.3.2 and their associated probability, so that for example $\tau(s_2, \texttt{car}_1?)(s_3, \texttt{change}!) = 0.1$ and $\tau(s_4, \texttt{car}_2?)(s_4, \texttt{stay}!) = 0.9$. ⧫

From this general definition of semi-Markov process, we can obtain the following standard models as special cases.

Generative: We get generative semi-Markov processes by letting $\texttt{In}$ be a singleton set. For a generative process, we will simply write $\tau(s)(s', a)$ for the transition function.

Reactive: Reactive semi-Markov processes, also known as semi-Markov decision processes, are obtained by letting $\texttt{In} = \texttt{Out}$, and in addition requiring that $\tau(s, a)(s', b) > 0$ only when $a = b$. For a reactive process, we write the transition function as $\tau(s, a)(s')$.

Continuous-time: If for every $s \in S$, $\rho(s)$ is an exponential distribution for some rate $\theta > 0$, then we obtain the popular model of continuous-time Markov chains.

Discrete-time: If we let $\rho(s) = \rho(s')$ for every $s, s' \in S$, then we obtain discrete-time Markov chains.

These can of course be combined to obtain e.g. continuous-time Markov decision processes, which are reactive continuous-time Markov chains. For the majority of this thesis, we will focus on the special cases of reactive and generative SMPs.

Remark 2.6.6. Continuous-time processes are often described in a different way than presented here. In the more common definition, there is no residence-time function, and instead there are rates given on the transitions. When adding actions on the transitions, this leads to the definition of (reactive) continuous-time Markov decision processes [8]. We recall here this alternative definition.

Definition 2.6.7 (Alternative)**.** A *continuous-time Markov decision process* is a pair $M = (S, \mathbf{R}, \ell)$, where S is a countable set of *states*, $\mathbf{R} : S \times \texttt{In} \times S \to \mathbb{R}_{\geq 0}$ is the *rate matrix*, and $\ell : S \to 2^{\mathcal{AP}}$ is the *labelling function*. ▲

We then have the following derived quantities. The *exit rate* of $s \in S$ under $a \in \mathtt{In}$ is given by $E(s,a) = \sum_{s' \in S} \mathbf{R}(s,a,s')$, and the *probability* of going from s to s' under a is given by $\mathbb{P}(s,a,s') = \frac{\mathbf{R}(s,a,s')}{E(s,a)}$.

Unfortunately, the two definitions are not equivalent. However, we can view our definition as a special case of the alternative definition by letting $\mathbf{R}(s,a,s') = \theta_s \cdot \tau(s,a)(s')$, where θ_s denotes the rate of the exponential distribution given by $\rho(s)$. Then we see that

$$E(s,a) = \sum_{s' \in S} \mathbf{R}(s,a,s') = \theta_s \cdot \sum_{s' \in S} \tau(s,a)(s') = \theta_s,$$

so that $E(s,a) = E(s,a')$ for all $a, a' \in \mathtt{In}$. We also get

$$\mathbb{P}(s,a,s') = \frac{\mathbf{R}(s,a,s')}{E(s,a)} = \tau(s,a)(s'),$$

as expected.

The reason that the two definitions are not equivalent is that in our definition, we must have $E(s,a) = E(s,a')$ for any $a, a' \in \mathtt{In}$, meaning that every action has the same exit rate, whereas in the alternative definition, each action can have a different exit rate. One could attempt to remedy this by modifying our definition to allow a different residence-time function ρ_a for each $a \in \mathtt{In}$, but we will not explore this idea further in this thesis.

Despite this difference from the more common definition, we have chosen the definition given in this thesis since it generalises more easily to semi-Markov processes. ⧫

Often we wish to construct systems by putting together smaller components. In order to do this, we need to define what it means to compose systems. Because our systems have real-time behaviour, in particular we need to describe how the real-time behaviour of the components influence the combined system. In order to accommodate different choices for combining real-time behaviour from the literature, we let this be described by a generic composition function.

Definition 2.6.8. A function $\star : \mathcal{D}(\mathbb{R}_{\geq 0}) \times \mathcal{D}(\mathbb{R}_{\geq 0}) \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is a *residence-time composition function* if it is commutative, meaning that

$$\star(\mu,\nu) = \star(\nu,\mu) \quad \text{for all } \mu,\nu \in \mathcal{D}(\mathbb{R}_{\geq 0}). \qquad \blacktriangle$$

For technical reasons, we will only consider composition of reactive SMPs.

Definition 2.6.9. Let $\star$ be a residence-time composition function. Then the $\star$-composition of $\mathcal{M}_1 = (S_1, \tau_1, \rho_1, \ell_1)$ and $\mathcal{M}_2 = (S_2, \tau_2, \rho_2, \ell_2)$, denoted $\mathcal{M}_1 \parallel_\star \mathcal{M}_2 = (S, \tau, \rho, \ell)$, is given by

1. $S = S_1 \times S_2$,
2. $\tau((s_1, s_2), a)((s_1', s_2')) = \tau_1(s_1, a)(s_1') \cdot \tau_2(s_2, a)(s_2')$,
3. $\rho((s_1, s_2)) = \star(\rho_1(s_1), \rho_2(s_2))$, and
4. $\ell((s_1, s_2)) = \ell(s_1) \cup \ell(s_2)$.

We will also write $s_1 \parallel_\star s_2$ to mean that $(s_1, s_2) \in S$. ▲

Similarly to the case of weighted transition systems, the standard notion of behavioural equivalence for SMPs is that of bisimulation.

Definition 2.6.10. Let $M = (S, \tau, \rho, \ell)$ be a SMP. A *bisimulation relation* is a relation $\mathcal{R} \subseteq S \times S$ such that $s_1 \mathcal{R} s_2$ implies

(B1) $\ell(s_1) = \ell(s_2)$,

(B2) $F_{s_1}(t) = F_{s_2}(t)$ for all $t \in \mathbb{R}_{\geq 0}$, and

(B3) for all $a \in \texttt{In}$ there exists a *weight function* $\Delta_a \in \mathcal{D}(S \times S \times \texttt{Out})$ such that

   (a) $\Delta_a(s, s', b) > 0$ implies $s \mathcal{R} s'$,
   (b) $\tau(s_1, a)(s, b) = \sum_{s' \in S} \Delta_a(s, s', b)$, and
   (c) $\tau(s_2, a)(s', b) = \sum_{s \in S} \Delta_a(s, s', b)$. ▲

The idea behind bisimulation for SMPs is the same as that for WTSs. Conditions (B1) and (B2) ensure that the information in the states is the same, by requiring that they have the same labels and the same residence-time function. Instead of the zig and zag conditions, we have the concept of a weight function in condition (B3). The weight function matches the probability mass of the transitions available to s_1 with the probability mass of the transitions available to s_2 in such a way that the bisimulation relation is preserved by the successor states. This means that when the probability mass of going from s_1 to a successor state s and outputting b, is matched with the probability mass of going from s_2 to s' and outputting b, then it must also hold that s and s' are in the bisimulation relation.

If there exists a bisimulation relation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$, then we will say that s_1 and s_2 are *bisimilar* and write $s_1 \sim s_2$. *Bisimilarity* is the largest bisimulation relation and is denoted by $\sim$.

A related notion is that of simulation. Whereas bisimulation guarantees that the processes have the same behaviour, simulation guarantees that one process can simulate any behaviour from the other process. Therefore, if s_1 simulates s_2, s_1 will be able to do anything that s_2 can do, but may also be able to do some things that s_2 can not do.

Definition 2.6.11. Let $\mathcal{M} = (S, \tau, \rho, \ell)$ be an SMP. A *simulation relation* is a relation $\mathcal{R} \subseteq S \times S$ such that $s_1 \mathcal{R} s_2$ implies

(S1) $\ell(s_1) = \ell(s_2)$,

(S2) $F_{s_1}(t) \leq F_{s_2}(t)$ for all $t \in \mathbb{R}_{\geq 0}$, and

(S3) for all $a \in \mathtt{In}$ there exists a *weight function* $\Delta_a \in \mathcal{D}(S \times S \times \mathtt{Out})$ such that

  (a) $\Delta_a(s, s', b) > 0$ implies $s \mathcal{R} s'$,

  (b) $\tau(s_1, a)(s, b) = \sum_{s' \in S} \Delta_a(s, s', b)$, and

  (c) $\tau(s_2, a)(s', b) = \sum_{s \in S} \Delta_a(s, s', b)$. ▲

The only difference between bisimulation and simulation is in conditions (B2) and (S2), where equality is used for bisimulation, whereas an inequality is used for simulation. This means that while bisimilarity is an equivalence relation, similarity is only a preorder.

If there exists a simulation relation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$, then we will say that s_2 *simulates* s_1 and write $s_1 \precsim s_2$. *Similarity* is the largest simulation relation and is denoted by $\precsim$.

Both simulation and bisimulation are at their core coinductive definitions, meaning that they compare elements step by step. However, sometimes we want to compare the entire history of an execution of a system with the execution of another system, rather than comparing them stepwise. We therefore also need to know what the probability of such an execution is. In order to define this probability, we first introduce the space of timed paths. The observable behaviour to keep track of in an execution of a SMP is the states that it visited, the time at which transitions were made, and the output actions that were performed. An *execution* or *path* is therefore an infinite sequence

$$\pi = (s_1, t_1, a_1), (s_2, t_2, a_2), (s_3, t_3, a_3), \cdots \in (S \times \mathbb{R}_{\geq 0} \times \mathtt{Out})^\omega.$$

Given a path π and $i \in \mathbb{N}$, we let

$$\pi[i] = s_i, \quad \pi\langle i \rangle = t_i, \quad \pi[\![i]\!] = a_i,$$

$$\pi|_i = (s_1, t_1, a_1), \ldots, (s_i, t_i, a_i), \quad \text{and} \quad \pi|^i = (s_i, t_i, a_i), (s_{i+1}, t_{i+1}, a_{i+1}), \ldots$$

We denote by $\Pi(M)$ the set of all paths in M and by

$$\Pi_n(M) = \{\pi|_n \mid \pi \in \Pi(M)\}$$

the set of all prefixes of length n of paths in M.

In order to turn $\Pi(M)$ into a measurable space, we need to construct a suitable σ-algebra. We will do this through the standard cylinder set construction. Given $n \geq 1$ and a set $E \subseteq \Pi_n(M)$, the *cylinder set* of rank n is the

set of all paths whose prefix up to the nth position agrees with that of E. The cylinder set of rank n is therefore given by

$$\mathfrak{C}(E) = \{\pi \in \Pi(M) \mid \pi|_n \in E\}.$$

This means that all paths in $\mathfrak{C}(E)$ begin exactly as prescribed by E, but after the nth step, they may start to differ. For notational convenience, given a set

$$E = S_1 \times O_1 \times R_1 \cdots \times S_n \times O_n \times R_n \subseteq \Pi_n(M),$$

we will often write $\mathfrak{C}(E)$ as

$$\mathfrak{C}(S_1 \dots S_n, O_1 \dots O_n, R_1 \dots R_n).$$

A cylinder $\mathfrak{C}(S_1 \dots S_n, O_1 \dots O_n, R_1 \dots R_n)$ is said to be *measurable* if $S_i \in 2^S$, $O_i \in 2^{\mathtt{Out}}$, and $R_i \in \mathbb{B}$ for all $1 \leq i \leq n$.

Lemma 2.6.12 ([2, Section 2.7])**.** *The set of measurable cylinders forms a Boolean algebra of sets.*

Definition 2.6.13. Let $M = (S, \tau, \rho, \ell)$ be a SMP. The *measurable space of paths* is the set of paths $\Pi(M)$ together with the σ-algebra Σ, defined as the smallest σ-algebra containing all measurable cylinders. ▲

Now that we have a σ-algebra for $\Pi(M)$, we wish to define a probability measure on paths. However, in order to do so, we must somehow resolve the non-determinism that is given by the input actions. In other words, we must decide on how the environment behaves when choosing inputs. This will be done by *schedulers*, which are also known in the literature as controllers, policies, or adversaries.

Definition 2.6.14. A *scheduler* is a function $\sigma : S^* \to \mathcal{D}(\mathtt{In})$. ▲

Intuitively, a scheduler looks at the history of visited states so far, and based on this information, it probabilistically chooses an input. One can also consider more complicated schedulers, such as schedulers that take into account the timed history [13], but we will not do so in this thesis.

Definition 2.6.15. Given a sequence of states $w = s_1 \dots s_k$ and a scheduler σ, we define the subprobability $\mathbb{P}^\sigma(w)$ inductively on measurable cylinders as

$$\mathbb{P}^\sigma(w)(\emptyset) = 0$$

$$\mathbb{P}^\sigma(w)(\mathfrak{C}(S, O, R)) = \rho(s_k)(R) \cdot \sum_{s' \in S} \sum_{a \in \mathtt{In}} \sum_{b \in O} \tau(s_k, a)(s', b) \cdot \sigma(w)(a), \text{ and}$$

$$\begin{aligned} &\mathbb{P}^\sigma(w)(\mathfrak{C}(S_1 S_2 \dots S_n, O_1 O_2 \dots O_n, R_1 R_2 \dots R_n)) \\ &= \rho(s_k)(R_1) \cdot \sum_{s' \in S_1} \sum_{a \in \mathtt{In}} \sum_{b \in O_1} \tau(s_k, a)(s', b) \cdot \sigma(w)(a) \\ &\quad \cdot \mathbb{P}^\sigma(ws')(\mathfrak{C}(S_2 \dots S_n, O_2 \dots O_n, R_2 \dots R_n)) \end{aligned}$$ ▲

For generative systems, $\mathtt{In}$ is a singleton, and therefore there is only one possible scheduler, namely the one that assigns probability one to the single element of $\mathtt{In}$ for any $w \in S^*$. When considering generative systems, we can therefore forget about schedulers.

Notice that we have only defined $\mathbb{P}^\sigma(w)$ on the Boolean algebra of measurable cylinders. We will now invoke the Hahn-Kolmogorov theorem to show that we can extend it to the measurable space of paths.

Lemma 2.6.16.

$$\begin{aligned}
&\mathbb{P}^\sigma(w)(\mathfrak{C}(S_1 \dots S_n, O_1 \dots O_n, R_1 \dots R_n)) \\
&= \sum_{s'_1 \in S_1} \sum_{a_1 \in \mathtt{In}} \sum_{b_1 \in O_1} \cdots \sum_{s_n \in S_n} \sum_{a_n \in \mathtt{In}} \sum_{b_n \in O_n} \tau(s_k, a_1)(s'_1, b_1) \cdots \tau(s'_{n-1}, a_n)(s'_n, b_n) \\
&\quad \cdot \sigma(w)(a_1) \cdots \sigma(w s'_1 \cdots s'_{n-1})(a_n) \\
&\quad \cdot \rho(s_k) \times \rho(s'_1) \times \cdots \times \rho(s'_{n-1})(R_1 \times \cdots \times R_n)
\end{aligned}$$

Proof. The result follows from unfolding the induction in Definition 2.6.15. ■

Theorem 2.6.17. *The subprobability $\mathbb{P}^\sigma(w)$ can be uniquely extended to a subprobability on the measurable space of paths.*

Proof. We will first argue that $\mathbb{P}^\sigma(w)$ is a pre-measure. Let Σ_0 denote the set of measurable cylinders. By Lemma 2.6.12, Σ_0 is a Boolean algebra. In order to show that $\mathbb{P}^\sigma(w)$ is a pre-measure, we therefore only need to show that conditions (P1) and (P2) from Definition 2.4.12 are satisfied. (P1) is satisfied by definition.

Next we consider condition (P2). For notational convenience, let

$$\bigcup_{m=1}^{\infty} \mathfrak{C}(S_{m,1} \dots S_{m,n_m}, O_{m,1} \dots O_{m,n_m}, R_{m,1} \dots R_{m,n_m}) = \bigcup_{m=1}^{\infty} \mathfrak{C}_m.$$

We must then show that

$$\mathbb{P}^\sigma(w)\left(\bigcup_{m=1}^{\infty} \mathfrak{C}_m\right) = \sum_{m=1}^{\infty} \mathbb{P}^\sigma(w)(\mathfrak{C}_m)$$

whenever $\bigcup_{m=1}^{\infty} \mathfrak{C}_m$ is a disjoint union of measurable cylinders such that $\bigcup_{m=1}^{\infty} \mathfrak{C}_m = \mathfrak{C}$ for some measurable cylinder

$$\mathfrak{C} = \mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n).$$

Note first that we can make all cylinders in $\bigcup_{m=1}^{\infty} \mathfrak{C}_m$ have the same length without affecting disjointness. This is because if

$$\mathfrak{C}(S_{i,1} \dots S_{i,n_i}, O_{i,1} \dots O_{i,n_i}, R_{i,1} \dots R_{i,n_i})$$

and

$$\mathfrak{C}(S_{j,1} \dots S_{j,n_j}, O_{j,1} \dots O_{j,n_j}, R_{j,1} \dots R_{j,n_j})$$

are two disjoint cylinders with $n_i < n_j$, then we can extend the first with

$$\begin{aligned}
&\mathfrak{C}(S_{i,1} \dots S_{i,n_i}, O_{i,1} \dots O_{i,n_i}, R_{i,1} \dots R_{i,n_i}) \\
&= \mathfrak{C}(S_{i,1} \dots S_{i,n_i} \underbrace{S \dots S}_{l \text{ times}}, O_{i,1} \dots O_{i,n_i} \underbrace{\texttt{Out} \dots \texttt{Out}}_{l \text{ times}}, R_{i,1} \dots R_{i,n_i} \underbrace{\mathbb{R} \dots \mathbb{R}}_{l \text{ times}})
\end{aligned}$$

where $l = n_j - n_i$, and the two cylinders are still disjoint. We can therefore assume without loss of generality that all the cylinders have length n. Let $w = s_1 \dots s_k$. By Lemma 2.6.16 we then get

$$\begin{aligned}
&\mathbb{P}^\sigma(w) \left(\bigcup_{m=1}^{\infty} \mathfrak{C}_m \right) \\
&= \mathbb{P}^\sigma(w)(\mathfrak{C}(S_1 \dots S_n, O_1 \dots O_n, R_1 \dots R_n) \\
&= \sum_{s_1' \in S_1} \sum_{a_1 \in \texttt{In}} \sum_{b_1 \in O_1} \cdots \sum_{s_n \in S_n} \sum_{a_n \in \texttt{In}} \sum_{b_n \in O_n} \tau(s_k, a_1)(s_1', b_1) \cdots \tau(s_{n-1}', a_n)(s_n', b_n) \\
&\quad \cdot \sigma(w)(a_1) \cdots \sigma(w s_1' \cdots s_{n-1}')(a_n) \\
&\quad \cdot \rho(s_k) \times \rho(s_1') \times \cdots \times \rho(s_{n-1}')(R_1 \times \cdots \times R_n) \\
&= \sum_{s_1' \in \bigcup_{m=1}^{\infty} S_{m,1}} \sum_{a_1 \in \texttt{In}} \sum_{b_1 \in \bigcup_{m=1}^{\infty} O_{m,1}} \cdots \sum_{s_n' \in \bigcup_{m=1}^{\infty} S_{m,n}} \sum_{a_n \in \texttt{In}} \sum_{b_n \in \bigcup_{m=1}^{\infty} O_{m,n}} \tau(s_k, a_1)(s_1', b_1) \\
&\quad \cdot \tau(s_1', a_2)(s_2', b_2) \cdots \tau(s_{n-1}', a_n)(s_n', b_n) \\
&\quad \cdot \sigma(w)(a_1) \cdots \sigma(w s_1' \cdots s_{n-1}')(a_n) \\
&\quad \cdot \rho(s_k) \times \rho(s_1') \times \cdots \times \rho(s_{n-1}') \left(\bigcup_{m=1}^{\infty} R_{m,1} \times \cdots \times \bigcup_{m=1}^{\infty} R_{m,n} \right) \\
&= \sum_{m=1}^{\infty} \sum_{s_1' \in S_{m,1}} \sum_{a_1 \in \texttt{In}} \sum_{b_1 \in O_{m,1}} \cdots \sum_{s_n' \in S_{m,n}} \sum_{a_n \in \texttt{In}} \sum_{b_n \in O_{m,n}} \tau(s_k, a_1)(s_1', b_1) \\
&\quad \cdot \tau(s_1', a_2)(s_2', b_2) \cdots \tau(s_{n-1}', a_n)(s_n', b_n) \\
&\quad \cdot \sigma(w)(a_1) \cdots \sigma(w s_1' \cdots s_{n-1}')(a_n) \\
&\quad \cdot \rho(s_k) \times \rho(s_1') \times \cdots \times \rho(s_{n-1}')(R_{m,1} \times \cdots \times R_{m,n}) \\
&= \sum_{m=1}^{\infty} \mathbb{P}^\sigma(w)(\mathfrak{C}_m),
\end{aligned}$$

and we conclude that condition (P2) is also satisfied.

We have thus shown that $\mathbb{P}^\sigma(w)$ is a pre-measure. Since the measurable space of paths is defined as the smallest σ-algebra containing all measurable cylinders, it then follows from the Hahn-Kolmogorov theorem (Theorem 2.4.13) that $\mathbb{P}^\sigma(w)$ can be uniquely extended to the measurable space of paths. ■

2.7 References

[1] "Complexity zoo," https://complexityzoo.uwaterloo.ca/Complexity_Zoo, accessed: 2018-05-16.

[2] R. B. Ash and C. A. Doléans-Dade, *Probability & Measure Theory*, 2nd ed. Harcourt/Academic Press, 1999.

[3] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[4] P. Billingsley, *Probability And Measure*, 3rd ed. Wiley-Interscience, 1995.

[5] P. Blackburn, J. F. A. K. van Benthem, and F. Wolter, *Handbook of Modal Logic*, ser. Studies in Logic and Practical Reasoning. Elsevier Science, 2006.

[6] S. Givant and P. Halmos, *Introduction to Boolean Algebras*, ser. Undergraduate Texts in Mathematics. Springer, 2009.

[7] J. Goubault-Larrecq, *Non-Hausdorff Topology and Domain Theory - Selected Topics in Point-Set Topology*, ser. New Mathematical Monographs. Cambridge University Press, 2013, vol. 22.

[8] M. R. Neuhäußer and J. Katoen, "Bisimulation and logical preservation for continuous-time Markov decision processes," in *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires and V. T. Vasconcelos, Eds., vol. 4703. Springer, 2007, pp. 412–427. [Online]. Available: https://doi.org/10.1007/978-3-540-74407-8

[9] P. Panangaden, *Labelled Markov Processes*. Imperial College Press, 2009.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 1994.

[11] M. Sipser, *Introduction to the Theory of Computation*, 2nd ed. Thomson course technology, 2006.

[12] T. Tao, *An Introduction to Measure Theory*, ser. Graduate studies in mathematics. American Mathematical Society, 2013.

[13] N. Wolovick and S. Johr, "A characterization of meaningful schedulers for continuous-time Markov decision processes," in *Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, September 25-27, 2006, Proceedings*,

ser. Lecture Notes in Computer Science, E. Asarin and P. Bouyer, Eds., vol. 4202. Springer, 2006, pp. 352–367. [Online]. Available: https://doi.org/10.1007/11867340_25

Chapter 3

Logical Specification Language for Reasoning About Bounds

This chapter summarises the content of Paper A "Reasoning About Bounds in Weighted Transition Systems" [6]. For the full paper, see Part II.

When using weighted transition systems (WTSs) as a modeling formalism, we encounter the approximate modeling problem: the quantities of interest may be irrational, whereas we can only ever measure rational values, and even then only with some uncertainty. It is therefore not clear which weight we should assign to each transition. Consider again the WTS in Example 1.3.1, where cleaning takes either 5, 10, or 15 minutes. These numbers are rather arbitrary, and we may easily imagine that cleaning could also take 6 minutes or 12.5 minutes. One way of approaching this problem is to allow *intervals* of transitions, and then reasoning about these. This is the approach taken by for example interval Markov chains [8] and interval weighted modal transition systems [9].

We will consider here a different, but related approach to the problem. Instead of reasoning about individual transitions or intervals of transitions, we reason about upper and lower bounds on the transitions. In Example 1.3.1, we may for instance say that cleaning takes at most 15 time units, so 15 is an upper bound on the time it takes to clean. Likewise, 5 is a lower bound. We argue that this is a reasonable point of view from an applications perspective, partly because bounds on quantities are easier to engineer than very precise measurements, and partly because many requirements that we are interested in verifying use upper and lower bounds, such as

> "the airbag must inflate within at most 2 milliseconds"

and

"traffic light must be green for at least 8 seconds before changing."

While the aforementioned approach of using intervals has some similarities with our approach of using bounds, namely that the endpoints of the intervals are also bounds, the two approaches are not equivalent. This is because when we talk about bounds, transitions with weights that are in between the bounds may or may not exist. To see this, consider again Example 1.3.1. 5 is a lower bound on transitions from the `cleaning` state to the `waiting state`, and 15 is an upper bound. However, the only transition that is allowed in between these bounds has weight 10. In contrast, interval approaches allow transitions with any weight that is within the interval.

Our focus is on logical aspects: we will introduce a logical specification language, which we call weighted logic with bounds (WLWB), and develop its metatheory, including a complete axiomatisation. We also argue that WLWB is the correct language for speaking about bounds in WTSs, by showing that it characterises a modified version of weighted bisimulation in which states are behaviourally equivalent when they have the same upper and lower bounds on behaviours. Furthermore, we give algorithms for solving the satisfiability and model checking problems for WLWB.

3.1 Weighted Logic With Bounds

First we introduce weighted logic with bounds (WLWB). We will denote formulas of WLWB by $\mathcal{L}$, and they are induced by the abstract syntax

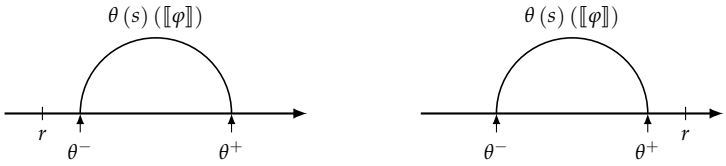
$$\mathcal{L}: \quad \varphi, \psi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid L_r\varphi \mid M_r\varphi$$

where $r \in \mathbb{Q}_{\geq 0}$ is a non-negative rational number and $p \in \mathcal{AP}$ is an atomic proposition. $\neg$ and $\wedge$ are the standard Boolean negation and conjunction, and the rest of the Boolean operators, such as disjunction and implication, can be derived from these.

The novel formulas are the ones of the form $L_r\varphi$ and $M_r\varphi$. Intuitively, $L_r\varphi$ says that it is possible to take a transition with weight *at least* r to a state where φ is true. Similarly, $M_r\varphi$ says that it is possible to take a transition with weight *at most* r to a state where φ is true.

In order to define the semantics of WLWB, we introduce some notation. Consider a WTS $\mathcal{M} = (S, \to, \ell)$. We then define the *image set* for a given state $s \in S$ and subset $T \subseteq S$ as

$$\theta(s)(T) = \{r \in \mathbb{R}_{\geq 0} \mid \text{there exists } t \in T \text{ such that } s \xrightarrow{r} t\}.$$



(a) $\mathcal{M}, s \models L_r\varphi$ **(b)** $\mathcal{M}, s \models M_r\varphi$

Figure 3.1.1: The semantics of L_r and M_r. If $\mathcal{M}, s \models L_r\varphi$, then r is to the left of $\theta^- (s) ([\![\varphi]\!])$, and if $\mathcal{M}, s \models M_r\varphi$, then r is to the right of $\theta^+ (s) ([\![\varphi]\!])$.

In other words $\theta(s)(T)$ is the set of all weights with which we can take a transition from s to some state in T. Furthermore, we let

$$\theta^- (s) (T) = \begin{cases} -\infty & \text{if } T = \emptyset \\ \inf \theta (s) (T) & \text{otherwise} \end{cases}$$

and

$$\theta^+ (s) (T) = \begin{cases} \infty & \text{if } T = \emptyset \\ \sup \theta (s) (T) & \text{otherwise,} \end{cases}$$

so $\theta^- (s) (T)$ is a lower bound on $\theta (s) (T)$ and $\theta^+ (s) (T)$ is an upper bound.

Given a WTS $\mathcal{M} = (S, \to, \ell)$, we now define the semantics of WLWB by the satisfaction relation $\models$ as follows.

$$\begin{array}{lll} \mathcal{M}, s \models p & \text{if and only if} & p \in \ell(s), \\ \mathcal{M}, s \models \neg\varphi & \text{if and only if} & \mathcal{M}, s \not\models \varphi, \\ \mathcal{M}, s \models \varphi \wedge \psi & \text{if and only if} & \mathcal{M}, s \models \varphi \text{ and } \mathcal{M}, s \models \psi, \\ \mathcal{M}, s \models L_r\varphi & \text{if and only if} & \theta^- (s) ([\![\varphi]\!]_{\mathcal{M}}) \geq r, \\ \mathcal{M}, s \models M_r\varphi & \text{if and only if} & \theta^+ (s) ([\![\varphi]\!]_{\mathcal{M}}) \leq r, \end{array}$$

where $[\![\varphi]\!]_{\mathcal{M}} = \{s \in S \mid \mathcal{M}, s \models \varphi\}$ is the set of all states of $\mathcal{M}$ that satisfies the formula φ. We will often omit the subscript $\mathcal{M}$ when it is clear which WTS is referred to.

The semantics of L_r and M_r is illustrated in Figure 3.1.1. The horizontal arrows represent the real number line, and the arches represent the part of the real number line in which elements of $\theta (s) ([\![\varphi]\!])$ may lie. The endpoints of the arches therefore correspond to $\theta^- (s) ([\![\varphi]\!])$ and $\theta^+ (s) ([\![\varphi]\!])$. This means that if a state satisfies $L_r\varphi$, then r must be to the left of the arch, and if a state satisfies $M_r\varphi$, then r must be to the right of the arch.

The operators L_r and M_r are inspired by similar operators in Markovian logic [4, 10], which in turn were inspired by logics for Harsanyi type spaces [1, 2]. However, although the intuition behind the operators are the same in both cases, they behave quite differently, since Markovian logic considers probabilities, whereas we consider weights, which have less structure

between them. We will see some of these differences when we consider axiomatisation in Section 3.3.

WLWB can express the usual "necessarily" and "possibly" operators from modal logic, written $\Box$ and $\Diamond$, respectively. To see this, note that $\Diamond\varphi$ means that it is possible to take a transition to where φ holds. Since all our weights are non-negative, $L_0\varphi$ is true if and only if it is possible to take a transition to where φ holds. The two are therefore equivalent, so we get

$$\Diamond\varphi = L_0\varphi \quad \text{and} \quad \Box\varphi = \neg L_0 \neg\varphi.$$

3.2 Bisimulation Using Bounds

We now argue that WLWB is the right language to reason about bounds in weighted transition systems. In order to do this, we define a new notion of bisimulation, which we call generalised weighted bisimulation. This notion of bisimulation only compares the upper and lower bounds of the possible behaviour of the systems. We then show that WLWB characterises exactly those states that are in a generalised weighted bisimulation relation with each other.

Definition 3.2.1. Let $\mathcal{M} = (S, \to, \ell)$ be a WTS. An equivalence relation $\mathcal{R} \subseteq S \times S$ is a *generalised weighted bisimulation relation* if for any states $s, t \in S$ we have that $s\mathcal{R}t$ implies

(Atomic harmony) $\ell(s) = \ell(t)$,

(Lower bound) $\theta^-(s)(T) = \theta^-(t)(T)$, and

(Upper bound) $\theta^+(s)(T) = \theta^-(t)(T)$

for any $\mathcal{R}$-equivalence class $T \subseteq S$. ▲

We will say that s and t are generalised weighted bisimilar and write $s \sim t$ if there exists a generalised weighted bisimulation relation $\mathcal{R}$ such that $s\mathcal{R}t$. Generalised weighted bisimilarity, denoted $\sim$, is the largest generalised weighted bisimulation relation.

Example 3.2.2. Consider the states s and t of the WTS in Figure 3.2.1. We will show that the relation

$$\mathcal{R} = \{(s,s), (t,t), (s',s'), (t',t'), (s,t), (t,s), (s',t'), (t',s')\}$$

is a generalised weighted bisimulation relation. It is clearly an equivalence relation.

For $(s',t') \in \mathcal{R}$, we have $\ell(s') = \{b\} = \ell(t')$, and since neither of the two states have any outgoing transitions, their lower and upper bounds also

Figure 3.2.1: $s \sim t$ but $s \not\sim_W t$.

match. For $(s,t) \in \mathcal{R}$, we have $\ell(s) = \{a\} = \ell(t)$, so atomic harmony is satisfied. For the bounds, note that $T = \{s', t'\}$ is the only equivalence class that can be reached from s and t, so this is the only equivalence class we need to consider. We have

$$\theta^- (s)(T) = \min\{1,2,3\} = 1 = \min\{1,3\} = \theta^- (t)(T)$$

and

$$\theta^+ (s)(T) = \max\{1,2,3\} = 3 = \max\{1,3\} = \theta^+ (t)(T),$$

so the lower and upper bounds also match. The remaining elements in the relation can be verified in a similar and symmetric manner. We therefore conclude that $s \sim t$.

On the other hand, it is not the case that $s \sim_W t$. To see this, simply note that $s \xrightarrow{2} s'$, which can not be matched by t, i.e. there is no state t'' such that $t \xrightarrow{2} t''$. ⧫

Example 3.2.2 also shows the essential difference between weighted bisimilarity and generalised weighted bisimilarity: weighted bisimilarity looks at all the individual transitions, whereas generalised weighted bisimilarity ignores the transitions in between the upper and lower bounds.

It is easy to see that if a relation is a weighted bisimulation, then it must also be a generalised weighted bisimulation, since if all transition weights match, then their lower and upper bounds must also match. Hence we get the following result, relating the two notions of bisimulation.

Theorem 3.2.3.

$$\sim_W \subseteq \sim \quad \textit{and} \quad \sim_W \neq \sim .$$

In order to prove the claim that WLWB characterises exactly those states that are generalised weighted bisimilar, we must restrict ourselves to a certain class of WTSs, namely those that are image-finite. The notion of image-finiteness is well-known in the literature, and is also necessary for other modal logics [3, 7].

Definition 3.2.4. A WTS $\mathcal{M} = (S, \to, \ell)$ is said to be *image-finite* if for any state $s \in S$ there are only finitely many states $t \in S$ such that $s \xrightarrow{r} t$ for some $r \in \mathbb{R}_{\geq 0}$. ▲

In other words, a WTS is image-finite if any state can only reach finitely many other states in one step.

Theorem 3.2.5. *For image-finite WTSs, we have*

$$s \sim t \quad \textit{if and only if} \quad \textit{for all } \varphi, s \models \varphi \textit{ if and only if } t \models \varphi.$$

3.3 Complete Axiomatisation

Having argued for why WLWB is an interesting logical specification language to consider, we now explore the properties of this language. As a first result, we will give a sound and complete axiomatisation of WLWB. Along the way, we will also obtain the finite model property.

The axiomatic system that we will consider is given by the axioms of propositional logic in addition to the axioms given in Table A.4.1. The axiom (A1) says that it is not possible to take a transition to where $\bot$ holds. Axioms (A2) and (A2′) give some monotonicity properties of L_r and M_r, whereas axioms (A3), (A3′), and (A4) show how L_r and M_r distribute over $\wedge$ and $\vee$. There is no axiom (A4′), which should be the obvious variant of (A4) with M_r instead of L_r. This is not because such an axiom would not be sound, but rather because it can be proved from the remaining axioms. Axioms (A5) and (A5′) say that if there is no transition to where ψ holds, then the upper and lower bounds for going to φ coincide with the upper and lower bounds for going to $\varphi \vee \psi$. Axioms (A6) and (A7) show how the L_r and M_r operators interact.

The rules (R1) and (R1′) say that if φ implies ψ, and we know that there is some transition to where φ holds, then an upper or lower bound for ψ is also an upper or lower bound for φ. Lastly, the rule (R2) says that if φ implies ψ, then if there is a transition to where φ holds, there must also be a transition to where ψ holds.

The axioms of Table 3.3.1 are sound, meaning that anything derived from the axioms must also be true semantically.

Theorem 3.3.1 (Soundness)**.**

$$\vdash \varphi \quad \textit{implies} \quad \models \varphi.$$

We will say that a formula φ is *consistent* if we can not derive $\bot$ from φ using the axioms. For a given consistent formula φ, we can construct a finite model $\mathcal{M}_\varphi$ with ultrafilters as states such that $\mathcal{M}_\varphi, s \models \varphi$ for some state s.

Theorem 3.3.2 (Finite model property)**.** *For any consistent formula $\varphi \in \mathcal{L}$, there exists a finite WTS $\mathcal{M} = (S, \to, \ell)$ and a state $s \in S$ such that $\mathcal{M}, s \models \varphi$.*

As an immediate consequence of the finite model property, we get that our axiomatisation is complete.

(A1):	$\vdash \neg L_0 \bot$	
(A2):	$\vdash L_{r+q}\varphi \to L_r\varphi$	if $q > 0$
(A2′):	$\vdash M_r\varphi \to M_{r+q}\varphi$	if $q > 0$
(A3):	$\vdash L_r\varphi \wedge L_q\psi \to L_{\min\{r,q\}}(\varphi \vee \psi)$	
(A3′):	$\vdash M_r\varphi \wedge M_q\psi \to M_{\max\{r,q\}}(\varphi \vee \psi)$	
(A4):	$\vdash L_r(\varphi \vee \psi) \to L_r\varphi \vee L_r\psi$	
(A5):	$\vdash \neg L_0\psi \to (L_r\varphi \to L_r(\varphi \vee \psi))$	
(A5′):	$\vdash \neg L_0\psi \to (M_r\varphi \to M_r(\varphi \vee \psi))$	
(A6):	$\vdash L_{r+q}\varphi \to \neg M_r\varphi$	if $q > 0$
(A7):	$\vdash M_r\varphi \to L_0\varphi$	
(R1):	$\vdash \varphi \to \psi \implies \vdash (L_r\psi \wedge L_0\varphi) \to L_r\varphi$	
(R1′):	$\vdash \varphi \to \psi \implies \vdash (M_r\psi \wedge L_0\varphi) \to M_r\varphi$	
(R2):	$\vdash \varphi \to \psi \implies \vdash L_0\varphi \to L_0\psi$	

Table 3.3.1: The axioms for our axiomatic system, where $\varphi, \psi \in \mathcal{L}$ and $q, r \in \mathbb{Q}_{\geq 0}$.

Theorem 3.3.3 (Completeness)**.**

$$\models \varphi \quad \textit{implies} \quad \vdash \varphi.$$

3.4 Satisfiability and Model Checking

Lastly we will consider some decision problems for WLWB. First we consider the problem of deciding whether a given formula is satisfiable, i.e. whether it has a model or not. In order to solve this problem, we will construct a tableau for a given formula φ. If the tableau is successful, meaning that it contains no inconsistencies, then we can construct a model for φ from the tableau. Otherwise, if the tableau is not successful, then we will know that there is no model for φ.

Given a formula φ, we start with the tuple $\langle \{\varphi\}, [0,0], [0,0] \rangle$ and then successively apply the rules of Table 3.4.1 until no further rules can be used. The (mod) rule may only be applied when no other rules can be applied.

The intuition behind a tuple $\langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle$, where Γ is a set of formulas and $\mathcal{I}^L$ and $\mathcal{I}^M$ are intervals, is that the current state must satisfy all the formulas in Γ, and any transitions to the current state must have a lower bound within the interval $\mathcal{I}^L$ and an upper bound within the interval $\mathcal{I}^M$. The (mod) rule signifies a state change, at which point all formulas in Γ have been broken down into either literals, i.e. formulas of the form p or $\neg p$ where $p \in \mathcal{AP}$, or modal formulas. The modal formulas then determine what transitions, if any, there must be from the current state to the next states.

$$(\wedge)\ \frac{\langle \Gamma \cup \{\varphi \wedge \psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\varphi, \psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\neg\wedge)\ \frac{\langle \Gamma \cup \{\neg(\varphi \wedge \psi)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\neg\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad \langle \Gamma \cup \{\neg\psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\neg\neg)\ \frac{\langle \Gamma \cup \{\neg\neg\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\text{mod})\ \frac{\langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1 \rangle \quad \cdots \quad \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k \rangle}$$

if $N^i \in \{L, M\}$ for all $1 \leq i \leq n$, $O^j \in \{L, M\}$ for all $1 \leq j \leq n'$, and no formula in Γ is of the form $N_r\varphi$ or $\neg N_r\varphi$ where $N \in \{L, M\}$.

Table 3.4.1: Tableau rules.

Example 3.4.1. We illustrate how the conclusions in the (mod) rule are constructed. Consider the tuple

$$\langle \{p_1, p_2, L_2p_1, L_4(p_1 \wedge p_2), L_0p_3, \neg L_5p_2, \neg M_6p_3\}, \mathcal{I}^L, \mathcal{I}^M \rangle.$$

By letting

$$\Gamma = \{p_1, p_2\}, \Gamma' = \{L_2p_1, L_4(p_1 \wedge p_2), L_0p_3\}, \text{ and } \Gamma'' = \{\neg L_5p_2, \neg M_6p_3\},$$

we get that $\langle \Gamma \cup \Gamma' \cup \Gamma'', \mathcal{I}^L, \mathcal{I}^M \rangle$ has the correct form for the hypothesis of the (mod) rule.

Now, the modal formulas in Γ' put requirements on the transitions to the next states. Consider the formulas L_2p_1 and $L_4(p_1 \wedge p_2)$. Any successor state where $p_1 \wedge p_2$ holds must also satisfy p_1. Hence we will not create two next states for these two formulas, but only one, which will be the most restrictive of the formulas, in this case $p_1 \wedge p_2$.

So we need two successor states, one that satisfies $p_1 \wedge p_2$ and one that satisfies p_3. It only remains to determine the weights on the transitions to these new states. For the state satisfying $p_1 \wedge p_2$, we see that the lower bounds must be at least 4, and the formula $\neg L_5p_2 \in \Gamma''$ tells us that the lower bound must be strictly less than 5. However, since no M_r formulas speak about p_1 or p_2, the upper bound has no restrictions. Hence the interval for the lower

bound is $\mathcal{I}^L = [4,5)$ and the interval for the upper bound is $\mathcal{I}^M = [0,\infty)$. Likewise we see that for the state satisfying p_3, the lower bound must be in the interval $[0,\infty)$, and the upper bound must be in the interval $[6,\infty)$.

Applying the mod rule to the tuple therefore gives the following result.

$$\text{(mod)}\ \frac{\langle\{p_1, p_2, L_2p_1, L_4(p_1 \wedge p_2), L_0p_3, \neg L_5p_2, \neg M_6p_3\}, \mathcal{I}^L, \mathcal{I}^M\rangle}{\langle\{p_1 \wedge p_2\}, [4,5), [0,\infty)\rangle \quad \langle\{p_3\}, [0,\infty)(6,\infty)\rangle} \qquad \blacklozenge$$

A tableau constructed from the tableau rules will be said to be *successful* if we can find a suitable subtree of the tableau such that for each tuple $\langle\Gamma, \mathcal{I}^L, \mathcal{I}^M\rangle$ in the subtree, the set of formulas Γ is consistent, and the intervals $\mathcal{I}^L$ and $\mathcal{I}^M$ are well-formed intervals.

The significance of the tableau construction follows from the next two lemmas.

Lemma 3.4.2. *φ is satisfiable if and only if there exists a successful tableau for φ.*

Lemma 3.4.3. *Given a successful tableau for φ, we can construct a model $\mathcal{M} = (S, \to, \ell)$ and a state $s \in S$ such that $\mathcal{M}, s \models \varphi$.*

This gives a decision procedure for the satisfiability problem: To determine whether φ is satisfiable, simply construct a tableau from the tableau rules of Table 3.4.1 starting with the tuple $\langle\{\varphi\}, [0,0], [0,0]\rangle$. If the tableau is successful, then φ is satisfiable, otherwise it is not satisfiable. Furthermore, if the tableau is successful, we can actually construct a model for φ from the successful tableau.

Theorem 3.4.4. *The satisfiability problem for WLWB is decidable.*

Example 3.4.5. Consider the formula $\varphi = \neg(\neg(L_2p_1 \wedge M_5L_1p_1) \wedge \neg M_2p_2))$. Using the tableau rules, we get the following tableau for φ.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\langle\{\neg(\neg(L_2p_1 \wedge M_5L_1p_1) \wedge M_2p_2)\}, [0,0], [0,0]\rangle}{\langle\{\neg\neg(L_2p_1 \wedge M_5L_1p_1)\}, [0,0], [0,0]\rangle}\,(\neg\wedge)}{\langle\{L_2p_1 \wedge M_5L_1p_1\}, [0,0], [0,0]\rangle}\,(\neg\neg)}{\langle\{L_2p_1, M_5L_1p_1\}, [0,0], [0,0]\rangle}\,(\wedge)}{\langle\{p_1, L_1p_1\}, [2,\infty), [5,\infty)\rangle}\,\text{(mod)}}{\langle\{p_1\}, [1,\infty), [0,\infty)\rangle}\,\text{(mod)} \qquad \cfrac{\cfrac{\langle\{\neg\neg M_2p_2\}, [0,0], [0,0]\rangle}{\langle\{M_2p_2\}, [0,0], [0,0]\rangle}\,(\neg\neg)}{\langle\{p_2\}, [0,\infty), [0,2]\rangle}\,\text{(mod)}$$

In this case the tableau is successful, every leaf and every node before a (mod) rule is consistent. From this tableau we can construct a model that satisfies φ. This model is shown in Figure 3.4.1 $\blacklozenge$

Lastly we consider the model checking problem for WLWB. This problem asks us to decide for a given model $\mathcal{M} = (S, \to, \ell)$, state $s \in S$, and formula φ whether $\mathcal{M}, s \models \varphi$.

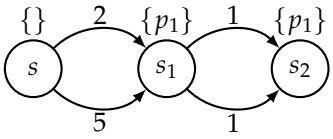


Figure 3.4.1: The model for the successful tableau in Example 3.4.5.

We can solve this problem in polynomial time by adapting the classical model checking algorithm by Clarke et al. [5] to our setting. This algorithm constructs a function F_φ, which assigns to each state the set of subformulas of φ that are true in that state. F_φ is built iteratively by first considering the smallest subformulas of φ (i.e. atomic propositions), then the second smallest subformulas (i.e. subformulas of the form $\neg p_1$, $p_1 \wedge p_2$, $L_r p_1$, or $M_r p_1$, where $p_1, p_2 \in \mathcal{AP}$), and so on until we get to φ itself. At each step we can use information about the smaller subformulas that have already been assigned by F_φ to determine which formulas must be assigned in the current step.

Lemma 3.4.6. *Given a model $\mathcal{M} = (S, \to, \ell)$, state $s \in S$, and formula φ, it holds that $\mathcal{M}, s \models \varphi'$ if and only if $\varphi' \in F_\varphi(s)$ for any subformula φ' of φ.*

By Lemma 3.4.6, we can therefore decide the model checking problem by checking whether $\varphi \in F_\varphi(s)$.

Theorem 3.4.7. *The model checking problem for WLWB is decidable in polynomial time.*

3.5 References

[1] R. J. Aumann, "Interactive epistemology I: knowledge," *Int. J. Game Theory*, vol. 28, no. 3, pp. 263–300, 1999. [Online]. Available: https://doi.org/10.1007/s001820050111

[2] ——, "Interactive epistemology II: probability," *Int. J. Game Theory*, vol. 28, no. 3, pp. 301–314, 1999. [Online]. Available: https://doi.org/10.1007/s001820050112

[3] P. Blackburn, J. F. A. K. van Benthem, and F. Wolter, *Handbook of Modal Logic*, ser. Studies in Logic and Practical Reasoning. Elsevier Science, 2006.

[4] L. Cardelli, K. G. Larsen, and R. Mardare, "Continuous Markovian logic - from complete axiomatization to the metric space of formulas," in *Computer Science Logic, 25th International Workshop / 20th Annual Conference of the EACSL, CSL 2011, September 12-15, 2011, Bergen, Norway,*

Proceedings, ser. LIPIcs, M. Bezem, Ed., vol. 12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011, pp. 144–158. [Online]. Available: http://dx.doi.org/10.4230/LIPIcs.CSL.2011.144

[5] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Trans. Program. Lang. Syst.*, vol. 8, no. 2, pp. 244–263, 1986. [Online]. Available: http://doi.acm.org/10.1145/5397.5399

[6] M. Hansen, K. G. Larsen, R. Mardare, and M. R. Pedersen, "Reasoning about bounds in weighted transition systems," *CoRR*, vol. abs/1703.03346, 2017. [Online]. Available: http://arxiv.org/abs/1703.03346

[7] M. Hennessy and R. Milner, "On observing nondeterminism and concurrency," in *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands, July 14-18, 1980, Proceedings*, ser. Lecture Notes in Computer Science, J. W. de Bakker and J. van Leeuwen, Eds., vol. 85. Springer, 1980, pp. 299–309. [Online]. Available: https://doi.org/10.1007/3-540-10003-2_79

[8] B. Jonsson and K. G. Larsen, "Specification and refinement of probabilistic processes," in *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*. IEEE Computer Society, 1991, pp. 266–277. [Online]. Available: https://doi.org/10.1109/LICS.1991.151651

[9] L. Juhl, K. G. Larsen, and J. Srba, "Modal transition systems with weight intervals," *J. Log. Algebr. Program.*, vol. 81, no. 4, pp. 408–421, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jlap.2012.03.008

[10] D. Kozen, R. Mardare, and P. Panangaden, "Strong completeness for Markovian logics," in *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and J. Sgall, Eds., vol. 8087. Springer, 2013, pp. 655–666. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40313-2_58

Chapter 4

Trace-Based Faster-Than Relation

This chapter summarises Paper B "Timed Comparisons of Semi-Markov Processes" [12] and paper C "A Faster-Than Relation for Semi-Markov Decision Processes" [11]. For the full papers, see Part II.

For real-time systems, non-functional requirements such as reliability, throughput, and response time are important to consider. It is therefore of interest to improve the worst-case timing guarantees on such systems, which leads us to investigate how to compare the timing behaviour of systems. In particular, we want to be able to describe when a system is faster than another, which will allow incremental timing improvements in a system.

Another important aspect of real-time systems is compositionality, which allows us to describe complex systems in terms of smaller components that together make up the whole system [3]. This leads to the picture in Figure 4.0.1, where we have a complex system consisting of $\mathcal{M}$ and the component $\mathcal{M}_2$, and we have a new component $\mathcal{M}_1$ which is faster than $\mathcal{M}_2$. The idea is then to replace $\mathcal{M}_2$ by $\mathcal{M}_1$ to obtain a faster system.

However, it is not always the case that replacing a slower component with a faster one leads to an overall system that is also faster. In other words, a local increase in timing behaviour may lead to a global decrease in timing behaviour. This is known as a *(parallel) timing anomaly* [8, 9]. We therefore also wish to investigate how to avoid such timing anomalies.

4.1 Faster-Than Relation

We first consider the question of what it means for one system to be faster than another. In this chapter, the systems we will consider are semi-Markov

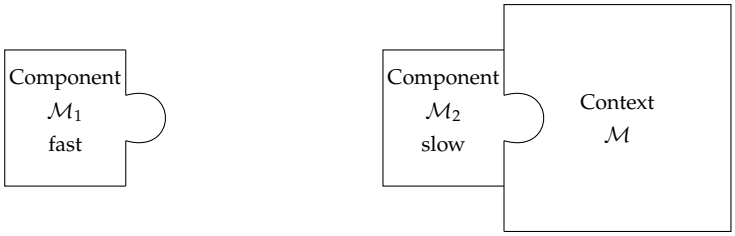


Figure 4.0.1: The context $\mathcal{M}$ operates in parallel with the component $\mathcal{M}_2$. If the component $\mathcal{M}_1$ is faster than $\mathcal{M}_2$, then if we replace $\mathcal{M}_2$ with $\mathcal{M}_1$, we would expect the overall behaviour to also be faster.

processes (SMPs), and we will take a trace-based view of SMPs. Intuitively, we will say that a state s_1 is faster than another state s_2 if any sequence of actions that s_2 can do within some time t', s_1 can do within some time $t \leq t'$ and with at least the same probability.

In order to do this, we must define the probability of completing a sequence of actions within a given time bound. Consider therefore an SMP $\mathcal{M} = (S, \tau, \rho, \ell)$, and recall from Section 2.6.2 that given n sets of states

$$S_1, \ldots S_n \in 2^S,$$

n sets of time points

$$R_1, \ldots, R_n \in \mathbb{B},$$

and n sets of output actions

$$O_1, \ldots, O_n \in 2^{\mathtt{Out}},$$

as well as a scheduler $\sigma : S^* \to \mathcal{D}(\mathtt{In})$, then

$$\mathbb{P}^\sigma(s)(\mathfrak{C}(S_1 \ldots S_n, O_1 \ldots O_n, R_1 \ldots R_n))$$

is the probability of starting in s, then going to a state $s_1 \in S_1$ within time $t_1 \in R_1$ while outputting $o_1 \in O_1$, then going from s_1 to a state in $s_2 \in S_2$ within time $t_2 \in R_2$ while outputting $o_2 \in O_2$, and so on.

Definition 4.1.1. Given a finite sequence of actions $o_1, \ldots, o_n \in \mathtt{Out}$ and a time bound $t \in \mathbb{R}_{\geq 0}$, we will say that

$$\mathfrak{C}(a_1 \ldots a_n, t) = \{\pi \in \Pi(\mathcal{M}) \mid \forall 1 \leq i \leq n, \pi[\![i]\!] = a_i \text{ and } \sum_{j=1}^{n} \pi\langle j \rangle \leq t\}$$

is a *time-bounded cylinder*. ▲

A time-bounded cylinder $\mathfrak{C}(a_1 \ldots a_n, t)$ therefore denotes all paths where the first n steps output the sequence $a_1 \ldots a_n$ and are done within time t. For

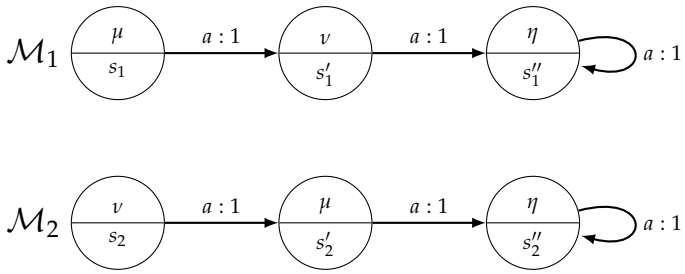


Figure 4.1.1: $\mathcal{M}_1$ is faster than $\mathcal{M}_2$.

example, the time-bounded cylinder $\mathfrak{C}(aa, 2)$ denotes that the first two output labels must be a's, and that the first two steps must be completed within 2 time units.

We will consider *pointed* SMPs, meaning that each SMP has a designated initial state. Given an SMP $\mathcal{M}$, we will write $(\mathcal{M}, s)$ to mean that s is the initial state of $\mathcal{M}$.

Definition 4.1.2. We will say that s_1 is *faster than* s_2 and write $s_1 \preceq s_2$ if for all schedulers σ there exists a scheduler σ' such that

$$\mathbb{P}^{\sigma'}(s_1)(C) \geq \mathbb{P}^{\sigma}(s_2)(C)$$

for all time-bounded cylinders C.

Given two pointed SMPs $(\mathcal{M}_1, s_1^*)$ and $(\mathcal{M}_2, s_2^*)$, we will say that $\mathcal{M}_1$ is *faster than* $\mathcal{M}_2$ and write $\mathcal{M}_1 \preceq \mathcal{M}_2$ if $s_1^* \preceq s_2^*$. ▲

Example 4.1.3. Consider the two SMPs $\mathcal{M}_1$ and $\mathcal{M}_2$ in Figure 4.1.1 and assume that $F_\mu(t) \geq F_\nu(t)$ for all $t \in \mathbb{R}_{\geq 0}$.

(Case $n = 1$) In this case we get

$$\mathbb{P}(s_1)(\mathfrak{C}(a, t)) = F_\mu(t) \quad \text{and} \quad \mathbb{P}(s_2)(\mathfrak{C}(a, t)) = F_\nu(t).$$

Since we have assumed $F_\mu(t) \geq F_\nu(t)$, it follows that

$$\mathbb{P}(s_1)(\mathfrak{C}(a, t)) \geq \mathbb{P}(s_2)(\mathfrak{C}(a, t)).$$

(Case $n > 1$) In this case we get

$$\mathbb{P}(s_1)(\mathfrak{C}(a^n, t)) = (\mu * \nu * \eta^{*(n-2)})([0, t])$$

and

$$\mathbb{P}(s_2)(\mathfrak{C}(a^n, t)) = (\nu * \mu * \eta^{*(n-2)})([0, t]),$$

where η^{*n} is the n-fold convolution of η. However, since convolution is commutative, it follows that

$$\mathbb{P}(s_1)(\mathfrak{C}(a^n, t)) = \mathbb{P}(s_2)(\mathfrak{C}(a^n, t)).$$

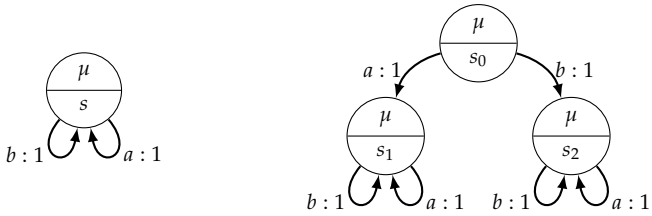


Figure 4.1.2: Example showing that the faster-than relation and the simulation relation are incomparable.

Thus we have $\mathbb{P}(s_1)(C) \geq \mathbb{P}(s_2)(C)$ for all time-bounded cylinders C, and therefore $s_1 \preceq s_2$. ♦

4.1.1 Comparison With Simulation and Bisimulation

In this section we compare our notion of a faster-than relation to the standard notions of simulation and bisimulation. For this, we also introduce the notion of two processes being equally fast.

Definition 4.1.4. $\mathcal{M}_1$ and $\mathcal{M}_2$ are *equally fast*, written $\mathcal{M}_1 \equiv \mathcal{M}_2$, if $\mathcal{M}_1 \preceq \mathcal{M}_2$ and $\mathcal{M}_2 \preceq \mathcal{M}_1$. ▲

Example 4.1.5. Consider the SMP in Figure 4.1.2 with the same probability measure μ in all states. It is easy to see that s is bisimilar to s_0, and hence s_0 also simulates s. However, we can show that $s \not\preceq s_0$ in the following way. Construct the (memoryless) scheduler σ by letting

$$\sigma(s_0)(a) = 0.5,\ \sigma(s_0)(b) = 0.5,\ \sigma(s_1)(a) = 1,\ \text{ and }\ \sigma(s_2)(b) = 1.$$

Now, for any scheduler σ', we must have either $\sigma'(s)(a) < 1$ or $\sigma'(s)(b) < 1$. If $\sigma'(s)(a) < 1$, then $\sigma'(s)(a) > (\sigma'(s)(a))^2 > \cdots > (\sigma'(s)(a))^n$. Furthermore, we see that

$$\mathbb{P}^{\sigma}(s_0)(\mathfrak{C}(a^n, t)) = 0.5 \cdot \mu^{*n}(t) \text{ and } \mathbb{P}^{\sigma'}(s)(\mathfrak{C}(a^n, t)) = (\sigma'(s)(a))^n \cdot \mu^{*n}(t)$$

for $n > 1$. Take some n such that $(\sigma'(s)(a))^n < 0.5$. In that case we get $\mathbb{P}^{\sigma'}(s)(\mathfrak{C}(a^n, t)) < \mathbb{P}^{\sigma}(s_0)(\mathfrak{C}(a^n, t))$. The same procedure can be used in case $\sigma'(s)(b) < 1$.

For schedulers with memory, notice that, starting from s, in each step either the probability of a trace consisting only of a's or the probability of a trace consisting only of b's must decrease. After some number of steps, the probability of one of these two must therefore decrease below 0.5, and then the rest of the argument is as before. Hence we conclude that $s \not\preceq s_0$, and therefore also that $s \not\equiv s_0$. ♦

Example 4.1.6. Consider again Figure 4.1.1 and let $F_\mu = Exp[\theta_1]$ and $F_\nu = Exp[\theta_2]$ with $\theta_1 > \theta_2 > 0$. Then, as shown in Example 4.1.3 we have $s_1 \preceq s_2$. However, we have both $s_1 \not\precsim s_2$ and $s_1 \not\sim s_2$. ♦

From Examples 4.1.5 and 4.1.6, we obtain the following theorem.

Theorem 4.1.7. *$\precsim$ and $\preceq$ are incomparable, $\sim$ and $\preceq$ are incomparable, and $\sim \subsetneq \equiv$.*

4.1.2 Algorithmic Considerations

When discussing algorithms for SMPs, we have to consider how residence-time distributions are handled by the algorithms. They must be described by some finite number of rational parameters that can be given as input to the algorithm, and since we are interested in the faster-than relation, we also want to be able to make comparisons between the distributions.

Definition 4.1.8. We say that a class of distributions $\mathcal{C}$ is *effective* if for any $\varepsilon > 0$, $b \in \mathbb{R}_{\geq 0}$, and $\mu, \nu \in \mathrm{Conv}(\mathcal{C})$,

$$\{t \in \mathbb{R}_{\geq 0} \mid \mu([0,t]) \geq \nu([0,t]) - \varepsilon \text{ and } t \leq b\}$$

is a semialgebraic set, where $\mathrm{Conv}(\mathcal{C})$ is the closure of $\mathcal{C}$ under convex combinations and convolutions. ▲

Semialgebraic sets are essentially those sets that can be described in the first-order theory of the reals, and since this theory is decidable [14], this allows us to compare distributions.

4.2 Hardness Results

In this section we consider the *faster-than problem*:

$$\text{Given } s_1 \text{ and } s_2, \text{ is it the case that } s_1 \preceq s_2?$$

Unfortunately, this problem turns out to be a difficult one. In particular, the faster-than problem is undecidable, and even approximating it is impossible.

In order to show these hardness results we rely on a connection to probabilistic automata [13]. A probabilistic automaton is a tuple

$$\mathcal{A} = (Q, A, q_0, \Delta : Q \times A \to \mathcal{D}_{=1}(Q), F),$$

where Q is a set of states, A is the alphabet, q_0 is the initial state, Δ is the transition function, and F is a set of accepting states. Many important problems for probabilistic automata are undecidable [5]. The problem that we will make use of is the *universality problem* for probabilistic automata which asks whether a given automaton $\mathcal{A}$ satisfies $\mathbb{P}_{\mathcal{A}}(w) \geq \frac{1}{2}$ for all words w.

Here, $\mathbb{P}_{\mathcal{A}}(w)$ is the probability for $\mathcal{A}$ to accept the word w. In other words, the universality problem asks whether all words are accepted with at least probability $\frac{1}{2}$. The universality problem is known to be undecidable [7, 10].

Given a probabilistic automaton $\mathcal{A}$, we construct the *derived* (generative) discrete-time Markov chain with output alphabet A

$$\mathcal{M}(\mathcal{A}) = (S, \tau, \ell),$$

where

- $S = (Q \times \{\ell, r\}) \cup \{\top\}$ for some new state $\top$,
- $\ell(s) = \emptyset$ for all $s \in S$, and
- τ is given by

$$\tau((p,\ell))((q,\ell),a) = \frac{1}{2|A|}\Delta(p,a)(q) \qquad \tau((p,\ell))(\top,a) = \frac{1}{2|A|} \text{ if } p \in F$$
$$\tau((p,r))((q,r),a) = \frac{1}{2|A|}\Delta(p,a)(q) \qquad \tau((p,r))(\top,a) = \frac{1}{4|A|}.$$

Now let $s_1 = (q_0, \ell)$ and $s_2 = (q_0, r)$ We then get

$$\mathbb{P}(s_1)(\mathfrak{C}(wa)) = \frac{1}{(2|A|)^{|w|+1}}(1 + \mathbb{P}_{\mathcal{A}}(w))$$

and

$$\mathbb{P}(s_2)(\mathfrak{C}(wa)) = \frac{1}{(2|A|)^{|w|+1}}\left(1 + \frac{1}{2}\right).$$

From this it follows that the faster-than problem for generative SMPs is undecidable, and a small extension of the argument shows that the same is true for reactive SMPs.

Theorem 4.2.1. *The faster-than problem is undecidable for both reactive and generative SMPs, and hence also for general SMPs.*

Note that the undecidability result does not depend on the real-time behaviour of the systems, since the derived Markov chain is discrete-time. This means that the difficulty with the faster-than problem is not actually the real-time behaviour of the systems, but rather the probabilistic branching structure.

We discuss three approaches to recover decidability:

- Imposing *structural restrictions* on the underlying graph,
- restricting the *observations* (i.e. input and output alphabet), and
- using *approximations*.

4.2.1 Structural Restrictions

The undecidability result for probabilistic automata already applies in the case of acyclic graphs, meaning that the only loops allowed are self-loops. Hence restricting to acyclic graphs will not help. However, there is another kind of structural restriction which has proved fruitful for probabilistic automata, which is that of *unambiguous* automata [6]. We will show in Section 4.4 that this notion also allows us to recover decidability for the faster-than problem in the case of generative systems.

4.2.2 Observations

The undecidability of the universality problem for probabilistic automata holds even when the alphabet only has two elements. Interestingly, the decidability of the universality problem is still an open problem when considering unary probabilistic automata, i.e. probabilistic automata where the alphabet only has a single symbol [2]. However, in this case the problem also has connections to the positivity problem for linear recurrence sequences [1], which has been a major open problem for decades. The positivity problem asks whether all terms of a given linear recurrence sequence are positive. It has been shown [1] that the universality problem is at least as hard as the positivity problem. Hence we get the following.

Theorem 4.2.2. *For generative processes with one output label, the faster-than problem is at least as hard as the positivity problem.*

Note that we have only been able to show the above for generative processes, since in that case the reduction from the universality problem works for only one symbol. However, in the reactive case the reduction we give requires us to introduce a new symbol, meaning that we need at least two symbols.

4.2.3 Approximations

By exploiting once again the connection to probabilistic automata, we can show that approximating the faster-than problem up to a multiplicative constant is impossible. This result relies on the following impossibility theorem for probabilistic automata.

Theorem 4.2.3 ([4, 5])**.** *Let $0 < \alpha < \beta < 1$ be two constants. There is no algorithm which, given a probabilistic automaton $\mathcal{A}$,*

- *returns **YES** if for all w we have $\mathbb{P}_{\mathcal{A}}(w) \geq \beta$ and*
- *returns **NO** if there exists w such that $\mathbb{P}_{\mathcal{A}}(w) \leq \alpha$.*

This in turn gives us the following impossibility result for approximating the faster-than problem.

Theorem 4.2.4. *Let $0 < \varepsilon < \frac{1}{3}$ be a constant. There is no algorithm which, given a discrete-time Markov chain $\mathcal{M}$ and two states s and s',*

- *returns **YES** if for all w we have $\mathbb{P}(s)(\mathfrak{C}(w)) \geq \mathbb{P}(s')(\mathfrak{C}(w))$ and*
- *returns **NO** if there exists w such that $\mathbb{P}(s)(\mathfrak{C}(w)) \leq \mathbb{P}(s')(\mathfrak{C}(w)) \cdot (1 - \varepsilon)$.*

However, in Section 4.3 we will show that approximation up to an additive constant can be done for a special kind of residence-time distributions, if we only consider what happens up to some given point in time.

4.3 Time-Bounded Additive Approximation

Although multiplicative approximation is impossible, we will now show that time-bounded additive approximation is possible. More precisely, we will show that the *time-bounded additive approximation problem* is decidable for a suitable class of residence-time distributions. This problem asks, given $\varepsilon > 0$, a time bound $b \in \mathbb{R}_{\geq 0}$, and two states s_1 and s_2, whether for all schedulers σ there exists a scheduler σ' such that

$$\mathbb{P}^{\sigma}(s_1)(C) \geq \mathbb{P}^{\sigma'}(s_2)(C) - \varepsilon \tag{4.1}$$

for all time-bounded cylinders $C = \mathfrak{C}(a_1 \dots a_n, t)$ where $t \leq b$.

Our decidability result holds for residence-time distributions that are *slow*. The formal definition of slow residence-time distributions is somewhat technical, but the idea is that slow residence-time distributions must use some non-zero amount of time to take a transition. This ensures that the process can not do infinitely many transitions within a given time bound, thus ruling out so-called Zeno behaviour. Furthermore, this means that the probability of time-bounded cylinders above some specific length must be less than $\varepsilon > 0$ for that given time bound, and hence the inequality in (4.1) is trivially satisfied. Therefore we only need to consider finitely many time-bounded cylinders.

Theorem 4.3.1. *Let $\mathcal{M}$ be an SMP with slow residence-time distributions. For any state s, $\varepsilon > 0$, time bound $b \in \mathbb{R}_{\geq 0}$, and scheduler σ, there exists $N \in \mathbb{N}$ such that*

$$\mathbb{P}^{\sigma}(s)(\mathfrak{C}(a_1 \dots a_n, b)) \leq \varepsilon \quad \text{for all } n \geq N.$$

Based on this result, we can prove the following theorem using the decidability of the first-order theory of the reals.

Theorem 4.3.2. *The time-bounded additive approximation problem is decidable for SMPs with effective and slow residence-time distributions.*

4.4 Unambiguous Processes

If we consider only generative processes, then we can also recover decidability by restricting to unambiguous processes. Intuitively, an unambiguous process is one in which a given output label uniquely identifies the successor state.

Definition 4.4.1. A generative SMP is *unambiguous* if for every $s \in S$ and $a \in \mathtt{Out}$ there exists at most one $s' \in S$ such that $\tau(s)(s', a) \neq 0$.

For an unambiguous SMP, we write $T(s, w)$ for the *unique* state reached from s after outputting the word w. ▲

Now consider the set of "loops" reachable from s and s', which we denote by $L(s_1, s_2) \subseteq S^2 \times \mathtt{Out}^{\leq S^2}$ and defined by

$$L(s_1, s_2) = \left\{ (p_1, p_2, v) \mid \exists w \in \mathtt{Out}^{\leq S^2}, \begin{array}{c} T(s_1, w) = p_1,\ T(s_2, w) = p_2, \\ T(p_1, v) = p_1,\ T(p_2, v) = p_2 \end{array} \right\}.$$

Intuitively, $(p_1, p_2, v) \in L(s_1, s_2)$ means that there exists a word w such that s_1 goes to p_1 and s_2 goes to p_2 when outputting w, and furthermore, whenever p_1 and p_2 output the word v, they end up back in p_1 and p_2. Thus, v makes p_1 and p_2 loop back to themselves. We can then prove the following lemma.

Lemma 4.4.2. *$s_1 \preceq s_2$ if and only if*

- *$\mathbb{P}(s_1)(\mathfrak{C}(w, t)) \geq \mathbb{P}(s_2)(\mathfrak{C}(w, t))$ for all $w \in \mathtt{Out}^{\leq S^2}$ and $t \in \mathbb{R}_{\geq 0}$ and*
- *$\mathbb{P}(p_1)(\mathfrak{C}(v, t)) \geq \mathbb{P}(p_2)(\mathfrak{C}(v, t))$ for all $(p_1, p_2, v) \in L(s_1, s_2)$ and $t \in \mathbb{R}_{\geq 0}$.*

Since for a given word, the inequalities in Lemma 4.4.2 can be checked for effective distributions, and since there are only finitely many words check, we obtain the following theorem.

Theorem 4.4.3. *For unambiguous generative SMPs with effective residence-time distributions, the faster-than problem is decidable in* **coNP**.

4.5 Logical Characterisation of the Faster-Than Relation

In this section we give a logical characterisation of the faster-than relation when we restrict to generative SMPs. The language $\mathcal{L}$ that we use for this consists of path formulas

$$\varphi ::= \top \mid \langle a \rangle \varphi$$

and state formulas

$$\psi ::= \mathcal{P}^{\leq t}_{\geq p}(\varphi)$$

where $t, p \in \mathbb{Q}_{\geq 0}$.

The semantics of $\mathcal{L}$ are given in terms of paths $\pi = a_1 a_2 \cdots \in \mathtt{Out}^*$ where we let $\pi[i] = a_i$ be the ith term of of π. Then the semantics of $\mathcal{L}$ are as follows.

$$\begin{array}{lll} \pi \models \top & & \text{always} \\ \pi \models \langle a \rangle \varphi & \text{iff} & \pi[1] = a \text{ and } \pi|_2 \models \varphi \\ s \models \mathcal{P}^{\leq t}_{\geq p}(\varphi) & \text{iff} & \mathbb{P}(s)(\mathfrak{C}(\mathfrak{W}(\varphi), t)) \geq p \end{array}$$

where $\pi|_2$ is the tail of π, and $\mathfrak{W}(\varphi)$ is the longest common prefix of all paths which satisfy φ.

The language $\mathcal{L}$ characterises the faster-than relation in the following sense.

Theorem 4.5.1. *For generative SMPs, it holds that*

$$s_1 \preceq s_2 \quad \textit{if and only if} \quad s_2 \models \psi \textit{ implies } s_1 \models \psi \textit{ for all } \psi \in \mathcal{L}.$$

Unfortunately, it is not clear to us how to make the logical characterisation work for the reactive case. The issue is that the definition of faster-than for the reactive case has an asymmetry in the quantifiers: For all schedulers σ there must exist a scheduler σ'. However, when defining the semantics of the operator, we must choose whether

$$\mathbb{P}^{\sigma}(s)(\mathfrak{C}(\mathfrak{W}(\varphi), t)) \geq p$$

should hold for all σ or just for some σ.

Apart from characterising the faster-than relation, the language $\mathcal{L}$ turns out to be quite simple. In particular, *every* formula in $\mathcal{L}$ is satisfiable, and even satisfiable by a finite model.

Theorem 4.5.2. *Any formula $\psi \in \mathcal{L}$ is satisfiable by a finite SMP.*

This can be easily seen by considering a path formula $\varphi = \langle a_1 \rangle \ldots \langle a_n \rangle \top$ and letting $\mathcal{M}_\varphi$ be an SMP with $n+1$ states such that state number i has an a_i-transition with probability 1 to state number $i+1$, and each state has a Dirac distribution at 0 as residence-time distribution.

As a corollary, this immediately implies that the satisfiability is trivially decidable.

Corollary 4.5.3. *The satisfiability problem for $\mathcal{L}$ is decidable.*

Finally, by making use of the existential theory of the reals, we obtain a **PSPACE** model checking algorithm for some commonly used residence-time distributions.

Theorem 4.5.4. *The model checking problem for $\mathcal{L}$ is decidable for SMPs with residence-time distributions that are either*

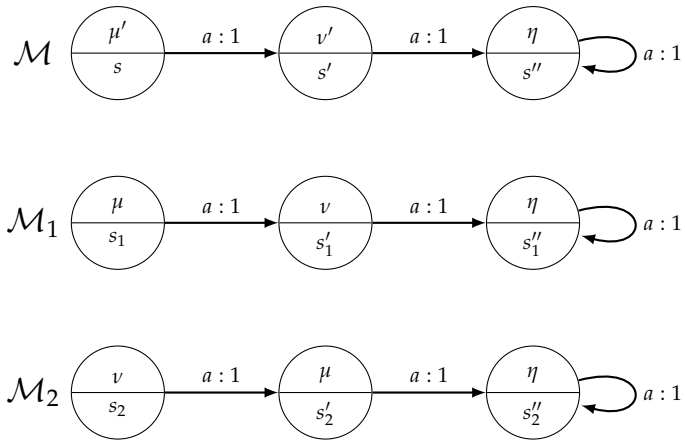


Figure 4.6.1: For different instantiations of μ, ν, μ', ν' and η', the context $\mathcal{M}$ together with the components $\mathcal{M}_1$ and $\mathcal{M}_2$ lead to parallel timing anomalies.

- *exponential,*
- *piecewise polynomial,*
- *piecewise affine,*
- *uniform, or*
- *Dirac.*

4.6 Compositionality

Let us now return to the picture in Figure 4.0.1, where a context $\mathcal{M}$ is operating in parallel with a component $\mathcal{M}_2$, and we wish to replace $\mathcal{M}_2$ by a faster component $\mathcal{M}_1$. We will first give some examples of when this picture can lead to parallel timing anomalies, meaning that even though $\mathcal{M}_1$ is faster than $\mathcal{M}_2$, the system $\mathcal{M}_1 \parallel_\star \mathcal{M}$ is actually slower than $\mathcal{M}_2 \parallel_\star \mathcal{M}$. We will give timing anomalies for the following types of composition function:

Product composition: $F_{\star(\mu,\nu)} = Exp[\theta \cdot \theta']$ if $F_\mu = Exp[\theta]$ and $F_\nu = Exp[\theta']$.

Minimum composition: $F_{\star(\mu,\nu)}(t) = \min\{F_\mu(t), F_\nu(t)\}$.

Maximum composition: $F_{\star(\mu,\nu)}(t) = \max\{F_\mu(t), F_\nu(t)\}$.

Consider the pointed SMPs $(\mathcal{M}, s)$, $(\mathcal{M}_1, s_1)$, and $(\mathcal{M}_2, s_2)$ as depicted in Figure 4.6.1, and let $F_\mu = Exp[2]$, $F_\nu = Exp[0.5]$, and $F_\eta = Exp[1]$. Notice that this immediately implies that $\mathcal{M}_1 \preceq \mathcal{M}_2$.

Example 4.6.1 (Product composition)**.** Let $F_{\mu'} = Exp[10]$ and $F_{\nu'} = Exp[0.1]$. We then get

$$\mathbb{P}(s_1 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.09$$

and

$$\mathbb{P}(s_2 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.30,$$

meaning that $\mathcal{M}_1 \parallel_\star \mathcal{M} \not\preceq \mathcal{M}_2 \parallel_\star \mathcal{M}$. ⧫

Example 4.6.2 (Minimum composition)**.** Let $F_{\mu'} = Exp[1]$ and $F_{\nu'} = Exp[2]$. We then get

$$\mathbb{P}(s_1 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.40$$

and

$$\mathbb{P}(s_2 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.51,$$

so also in this case $\mathcal{M}_1 \parallel_\star \mathcal{M} \not\preceq \mathcal{M}_2 \parallel_\star \mathcal{M}$. ⧫

Example 4.6.3 (Maximum composition)**.** Let $F_{\mu'} = Exp[2]$ and $F_{\nu'} = Exp[1]$. We then get

$$\mathbb{P}(s_1 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.75$$

and

$$\mathbb{P}(s_2 \parallel_\star s)(\mathfrak{C}(aa, 2)) \approx 0.91,$$

so once again we get $\mathcal{M}_1 \parallel_\star \mathcal{M} \not\preceq \mathcal{M}_2 \parallel_\star \mathcal{M}$. ⧫

4.6.1 Avoiding Parallel Timing Anomalies

We have now seen that parallel timing anomalies can occur for many standard ways of composing systems. Furthermore, none of the examples we showed made use of non-determinism or probabilistic branching, showing that parallel timing anomalies can occur purely as a consequence of the real-time behaviour of the systems.

We therefore wish to understand under which conditions we can ensure that parallel timing anomalies do not occur. We provide a first step toward such an understanding by identifying a set of conditions which over-approximate the faster-than relation, and show that this set of conditions is decidable. Hence, we can algorithmically verify that a system satisfies the conditions, and give guarantees that the system can not lead to parallel timing anomalies.

In order to over-approximate the faster-than relation, we require that $\mathcal{M}_1 \parallel_\star \mathcal{M}$ is pointwise faster than $\mathcal{M}_1$ along all paths (from the initial states). Likewise, we require that $\mathcal{M}_2$ is pointwise faster than $\mathcal{M}_2 \parallel_\star \mathcal{M}$ along all paths. Since we already know that $\mathcal{M}_1$ is faster than $\mathcal{M}_2$, this will imply by transitivity that $\mathcal{M}_1 \parallel_\star \mathcal{M}$ is faster than $\mathcal{M}_2 \parallel_\star \mathcal{M}$, thus ensuring that parallel timing anomalies can not occur.

Definition 4.6.4. An reactive SMP $\mathcal{M} = (S, \tau, \rho, \ell)$ has a *deterministic Markov kernel* if for all $s \in S$ and $a \in \mathtt{In}$ there is at most one state $s' \in S$ such that $\tau(s,a)(s') > 0$. ▲

Definition 4.6.5. Let $n \in \mathbb{N}$. We say that $\star$ is *n-monotonic* in $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}$, and $\mathcal{M}'$ and write $(\mathcal{M}_1, \mathcal{M}) \lessapprox^n_\star (\mathcal{M}_2, \mathcal{M}')$ if $\mathcal{M}'$ has a deterministic Markov kernel and the following holds pointwise along all paths of length up to n:

1. The CDF of $\mathcal{M}_1 \parallel_\star \mathcal{M}$ is pointwise greater than that of $\mathcal{M}$.
2. The CDF of $\mathcal{M}_2$ is pointwise greater than that of $\mathcal{M}_2 \parallel_\star \mathcal{M}'$.
3. For all schedulers σ there exists a scheduler σ' such that the transition probability of $\mathcal{M}_1 \parallel_\star \mathcal{M}$ under σ' is greater than that of $\mathcal{M}$ under σ.
4. For all schedulers σ there exists a scheduler σ' such that the transition probability of $\mathcal{M}_2$ under σ' is greater than that of $\mathcal{M}_2 \parallel_\star \mathcal{M}'$ under σ.

We will say that $\star$ is *monotonic* in $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}$, and $\mathcal{M}'$ and write $(\mathcal{M}_1, \mathcal{M}) \lessapprox_\star (\mathcal{M}_2, \mathcal{M}')$ if $(\mathcal{M}_1, \mathcal{M}) \lessapprox^n_\star (\mathcal{M}_2, \mathcal{M}')$ for all n. ▲

Theorem 4.6.6. *If $\mathcal{M}_1 \preceq \mathcal{M}_2$, $\mathcal{M} \preceq \mathcal{M}'$, and $(\mathcal{M}_1, \mathcal{M}) \lessapprox_\star (\mathcal{M}_2, \mathcal{M}')$, then $\mathcal{M}_1 \parallel_\star \mathcal{M} \preceq \mathcal{M}_2 \parallel_\star \mathcal{M}'$.*

We do not know whether the conditions of monotonicity are decidable, but if we strength the existential quantifiers of items 3 and 4 in Definition 4.6.5 to universal quantifiers, then we arrive at a notion of *strong* n-monotonicity, respectively *strong* monotonicity, denoted by $(\mathcal{M}_1, \mathcal{M}) \leqq^n_\star (\mathcal{M}_2, \mathcal{M}')$, respectively $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$.

Since clearly $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$ implies $(\mathcal{M}_1, \mathcal{M}) \lessapprox_\star (\mathcal{M}_1, \mathcal{M}')$, we get the following corollary.

Corollary 4.6.7. *If $\mathcal{M}_1 \preceq \mathcal{M}_2$, $\mathcal{M} \preceq \mathcal{M}'$, and $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$, then $\mathcal{M}_1 \parallel_\star \mathcal{M} \preceq \mathcal{M}_2 \parallel_\star \mathcal{M}'$.*

The first step in order to decide whether $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$ is to notice that it is enough to consider paths up to a specific length for finite systems.

Lemma 4.6.8. *Let $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}$, and $\mathcal{M}'$ be finite SMPs and let*

$$m = \max\{|\mathcal{M}_1| \cdot |\mathcal{M}|, |\mathcal{M}_2| \cdot |\mathcal{M}'|\} + \max\{|\mathcal{M}_1|, |\mathcal{M}_2|, |\mathcal{M}|, |\mathcal{M}'|\} + 1.$$

If $(\mathcal{M}_1, \mathcal{M}) \leqq^m_\star (\mathcal{M}_2, \mathcal{M}')$, then $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$.

We can then make use of the first-order theory of the reals to decide strong monotonicity.

Theorem 4.6.9. *Let $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}$, and $\mathcal{M}'$ be finite, reactive SMPs. If for all paths π_1 in $\mathcal{M}_1 \parallel_\star \mathcal{M}$, π_2 in $\mathcal{M}$, π_3 in $\mathcal{M}_2$, and π_4 in $\mathcal{M}_2 \parallel_\star \mathcal{M}'$, the sets*

$$\{t \in \mathbb{R}_{\geq 0} \mid F_{\pi_1[i]}(t) \geq F_{\pi_2[i]}\}$$

and

$$\{t \in \mathbb{R}_{\geq 0} \mid F_{\pi_3[i]}(t) \geq F_{\pi_4[i]}\}$$

are semialgebraic for all $1 \leq i \leq m$, then it is decidable whether $(\mathcal{M}_1, \mathcal{M}) \leqq_\star (\mathcal{M}_2, \mathcal{M}')$.

The sets described in Theorem 4.6.9 are semialgebraic for common distributions such as exponential and uniform distributions, and for common composition functions such as product, minimum, and maximum composition.

4.7 References

[1] S. Akshay, T. Antonopoulos, J. Ouaknine, and J. Worrell, "Reachability problems for Markov chains," *Inf. Process. Lett.*, vol. 115, no. 2, pp. 155–158, 2015.

[2] S. Akshay, B. Genest, B. Karelovic, and N. Vyas, "On regularity of unary probabilistic automata," in *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, ser. LIPIcs, N. Ollinger and H. Vollmer, Eds., vol. 47. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 8:1–8:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2016.8

[3] E. M. Clarke, D. E. Long, and K. L. McMillan, "Compositional model checking," in *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 1989, pp. 353–362. [Online]. Available: https://doi.org/10.1109/LICS.1989.39190

[4] A. Condon and R. J. Lipton, "On the complexity of space bounded interactive proofs (extended abstract)," in *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*. IEEE Computer Society, 1989, pp. 462–467. [Online]. Available: https://doi.org/10.1109/SFCS.1989.63519

[5] N. Fijalkow, "Undecidability results for probabilistic automata," *SIGLOG News*, vol. 4, no. 4, pp. 10–17, 2017. [Online]. Available: http://doi.acm.org/10.1145/3157831.3157833

[6] N. Fijalkow, C. Riveros, and J. Worrell, "Probabilistic automata of bounded ambiguity," in *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, ser. LIPIcs, R. Meyer and U. Nestmann, Eds., vol. 85. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, pp. 19:1–19:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.CONCUR.2017.19

[7] H. Gimbert and Y. Oualhadj, "Probabilistic automata on finite words: Decidable and undecidable problems," in *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, ser. Lecture Notes in Computer Science, S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, Eds., vol. 6199. Springer, 2010, pp. 527–538. [Online]. Available: https://doi.org/10.1007/978-3-642-14162-1_44

[8] R. Kirner, A. Kadlec, and P. P. Puschner, "Precise worst-case execution time analysis for processors with timing anomalies," in *21st Euromicro Conference on Real-Time Systems, ECRTS 2009, Dublin, Ireland, July 1-3, 2009*. IEEE Computer Society, 2009, pp. 119–128. [Online]. Available: https://doi.org/10.1109/ECRTS.2009.8

[9] T. Lundqvist and P. Stenström, "Timing anomalies in dynamically scheduled microprocessors," in *Proceedings of the 20th IEEE Real-Time Systems Symposium, Phoenix, AZ, USA, December 1-3, 1999*. IEEE Computer Society, 1999, pp. 12–21. [Online]. Available: https://doi.org/10.1109/REAL.1999.818824

[10] A. Paz, *Introduction to Probabilistic Automata*. Academic Press, 1971.

[11] M. R. Pedersen, G. Bacci, and K. G. Larsen, "A faster-than relation for semi-Markov decision processes," *CoRR*, vol. abs/1810.11243, 2018. [Online]. Available: https://arxiv.org/abs/1810.11243

[12] M. R. Pedersen, N. Fijalkow, G. Bacci, K. G. Larsen, and R. Mardare, "Timed comparisons of semi-Markov processes," in *Language and Automata Theory and Applications - 12th International Conference, LATA 2018, Ramat Gan, Israel, April 9-11, 2018, Proceedings*, ser. Lecture Notes in Computer Science, S. T. Klein, C. Martín-Vide, and D. Shapira, Eds., vol. 10792. Springer, 2018, pp. 271–283. [Online]. Available: https://doi.org/10.1007/978-3-319-77313-1_21

[13] M. O. Rabin, "Probabilistic automata," *Information and Control*, vol. 6, no. 3, pp. 230–245, 1963. [Online]. Available: https://doi.org/10.1016/S0019-9958(63)90290-0

[14] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*. University of California press, 1951.

Chapter 5

Simulation-Based Faster-Than Relation

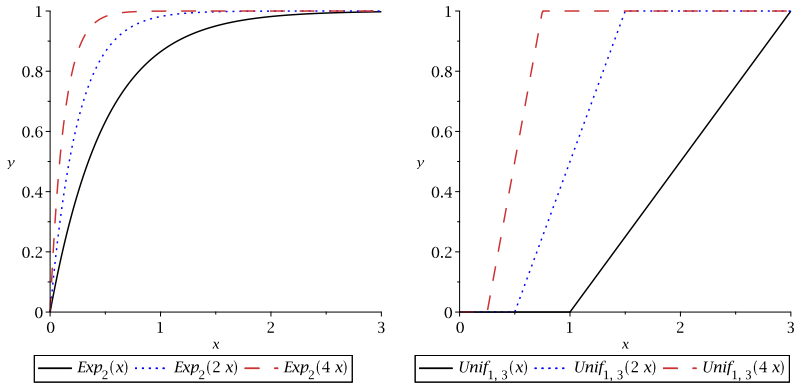
This chapter summarises Paper D "A Hemimetric Extension of Simulation for Semi-Markov Decision Processes" [3]. For the full paper, see Part II.

We have seen in Chapter 4 how to compare the real-time behaviour of SMPs by considering their traces. In this chapter we will instead compare processes through the notion of *simulation*. Roughly speaking, a process s_1 simulates a process s_2 if anything that s_2 can do, s_1 can also do. However, s_1 may be able to do more than what s_2 can do. When considering the real-time behaviour of systems, we in addition require that s_1 must be faster than s_2 in order for s_1 to simulate s_2.

Since the real-time behaviour of processes is sensitive to the exact type of distribution and parameters used to specify the real-time behaviour in each state, such processes are subject to the approximate modeling problem, which we discussed in the introduction. We therefore develop a notion of *simulation distance*, which quantifies how close a process is to simulating another process in terms of its real-time behaviour. In order to do this, we first consider how to quantitatively compare the residence-time distributions of processes. In this chapter, we only consider *reactive* processes.

5.1 Comparing Residence-Time Distributions

In order to compare the residence-time distributions of processes, we will take as our starting point the *usual stochastic order* from the theory of stochastic orders [4]. A distribution μ is smaller than another distribution ν in the usual stochastic order if the CDF of μ is point-wise greater than the CDF of



(a) Exponential **(b)** Uniform

Figure 5.1.1: When accelerating by a factor $\varepsilon \geq 1$, the CDF becomes faster. Here the CDFs of an exponential and a uniform distribution are plotted where the acceleration factor ε takes the values 1, 2, and 4.

ν, i.e. when

$$F_\mu(t) \geq F_\nu(t) \quad \text{for all } t \in \mathbb{R}_{\geq 0}.$$

We extend this order to a quantitative notion, and while doing so, we also shift the focus from the distributions to the CDFs, since we are only interested in comparing CDFs.

Definition 5.1.1. Let F and G be CDFs and let $\varepsilon \in \mathbb{R}_{>0}$. We say that F is *ε-faster than* G and write $F \sqsubseteq_\varepsilon G$ if

$$F(\varepsilon \cdot t) \geq G(t) \quad \text{for all } t \in \mathbb{R}_{\geq 0}. \qquad \blacktriangle$$

The name ε-faster-than comes from the fact that if $F \sqsubseteq_\varepsilon G$, then at every point in time t, F will have a higher probability of having fired a transition than G, if we accelerate the real-time behaviour of F by the factor ε.

Example 5.1.2. To get a feeling for the significance of the acceleration factor ε, consider the plots in Figure 5.1.1.

In Figure 5.1.1a, we see the CDF of an exponential distribution with rate 2. When accelerating this CDF by a factor 2, we see that the shape of an exponential distribution is preserved, but the resulting CDF is faster. The same happens with acceleration factor 4, except the resulting CDF is even faster. Thus the net result of accelerating an exponential distribution is to increase its rate.

In Figure 5.1.1b, we see the CDF of a uniform distribution between 1 and 3. When accelerating this CDF by a factor 2, the shape of a uniform distribution is preserved, but the resulting uniform distribution is between two

values that are less than the original values, thus making the CDF faster. The same happens with acceleration factor 4, except now the uniform distribution is between values that are even smaller, resulting in an even faster CDF. The result of accelerating a uniform distribution is therefore to decrease the parameters of the uniform distribution. ⧫

The next three propositions show for what values of ε the ε-faster-than relation holds between Dirac, uniform, and exponential distributions.

Proposition 5.1.3. *Let F be any CDF. The following holds for any $\varepsilon \in \mathbb{R}_{>0}$.*

1. *$Dirac[0] \sqsubseteq_\varepsilon F$.*
2. *If $F \neq Dirac[0]$, then $F \not\sqsubseteq_\varepsilon Dirac[0]$.*

Proposition 5.1.4.

1. *$Exp[\theta_1] \sqsubseteq_\varepsilon Exp[\theta_2]$, where $\varepsilon = \frac{\theta_2}{\theta_1}$.*
2. *If $c = 0$ and $a > 0$, then $Unif[a,b] \not\sqsubseteq_\varepsilon Unif[c,d]$ for any $\varepsilon \in \mathbb{R}_{>0}$.*
3. *If $c = 0$ and $a = 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Unif[c,d]$, where $\varepsilon = \frac{b}{d}$.*
4. *If $c > 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Unif[c,d]$, where $\varepsilon = \max\left\{\frac{a}{c}, \frac{b}{d}\right\}$.*

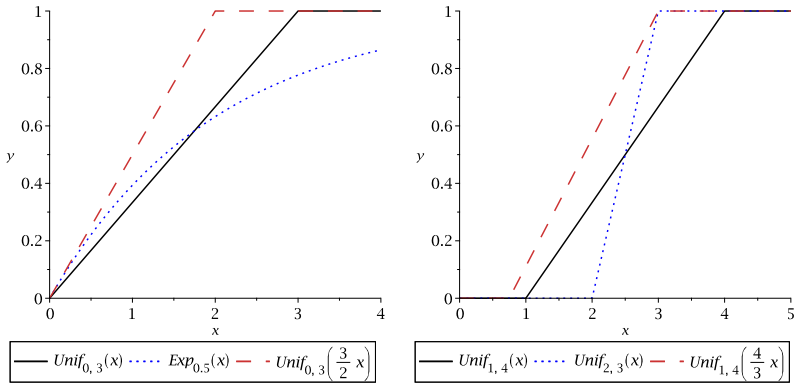
In all cases, the given ε is the least such that the ε-faster than relation holds.

Proposition 5.1.5.

1. *$Exp[\theta] \not\sqsubseteq_\varepsilon Unif[a,b]$ for all $\varepsilon \in \mathbb{R}_{>0}$.*
2. *If $a > 0$, then $Unif[a,b] \not\sqsubseteq_\varepsilon Exp[\theta]$ for all $\varepsilon \in \mathbb{R}_{>0}$.*
3. *If $a = 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Exp[\theta]$, where $\varepsilon = \theta \cdot b$. Furthermore, this is the least ε such that the ε-faster-than relation holds.*

Example 5.1.6. Consider the plots in Figure 5.1.2. In Figure 5.1.2a, we see an exponential distribution with rate 0.5 and a uniform distribution between 0 and 3. Clearly, neither of them is faster than the other, since they cross. However, Proposition 5.1.5 tells us that if we accelerate the uniform distribution by a factor $\varepsilon = 0.5 \cdot 3 = 1.5$, then the resulting uniform distribution is faster than the exponential distribution, and this can also be seen in Figure 5.1.2a.

In Figure 5.1.2b we see two different uniform distributions, one between 1 and 4 and another between 2 and 3. Again, neither of these is faster than the other, but by Proposition 5.1.4, accelerating the uniform distribution between 1 and 4 by a factor $\varepsilon = \max\{\frac{1}{3}, \frac{4}{3}\} = \frac{4}{3}$ results in a uniform distribution that is faster than the one between 2 and 3, as can be seen in Figure 5.1.2b. ⧫



(a) Exponential **(b)** Uniform

Figure 5.1.2: Accelerating the real-time behaviour of a uniform distribution to make it faster than another uniform distribution or an exponential distribution.

In addition, the ε-faster-than relation enjoys the following properties. The first property is that the relation is monotonic in ε, and the second property is that the relation is a congruence with respect to convolution.

Lemma 5.1.7. *Let $\varepsilon \leq \varepsilon'$ and assume that $F \sqsubseteq_\varepsilon G$. Then $F \sqsubseteq_{\varepsilon'} G$.*

Lemma 5.1.8. *If $F_{\mu_1} \sqsubseteq_\varepsilon F_{\mu_2}$ and $F_{\nu_1} \sqsubseteq_\varepsilon F_{\nu_2}$, then $F_{(\mu_1 * \nu_1)} \sqsubseteq_\varepsilon F_{(\mu_2 * \nu_2)}$.*

5.2 Simulation Distance

We will now use the ε-faster-than relation to extend simulation for SMPs to a distance between SMPs, which intuitively measures how much one process needs to be accelerated in order to simulate another process. To do this, first note that condition (S2) in Definition 2.6.11 of simulation for SMPs is actually an instance of the usual stochastic order. We can therefore naturally extend this definition using our notion of ε-faster-than.

Definition 5.2.1. Let $\mathcal{M} = (S, \tau, \rho, \ell)$ be an SMP. A *ε-simulation relation* is a relation $\mathcal{R} \subseteq S \times S$ such that $s_1 \mathcal{R} s_2$ implies

1. $\ell(s_1) = \ell(s_2)$,
2. $F_{s_2} \sqsubseteq_\varepsilon F_{s_1}$, and
3. for all $a \in \mathtt{In}$ there exists a *weight function* $\Delta_a \in \mathcal{D}(S \times S)$ such that
   (a) $\Delta_a(s, s') > 0$ implies $s \mathcal{R} s'$,
   (b) $\tau(s_1, a)(s) = \sum_{s' \in S} \Delta_a(s, s')$, and

Figure 5.2.1: Two states of an SMP $\mathcal{M}$.

(c) $\tau(s_2, a)(s') = \sum_{s \in S} \Delta_a(s, s')$.

If there exists an ε-simulation relation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$, then we say that s_2 *ε-simulates* s_1 and write $s_1 \precsim_\varepsilon s_2$. ▲

Example 5.2.2. Consider the two states in Figure 5.2.1. Since both states have exponential residence-time distributions, Proposition 5.1.4 tells us that

$$s_1 \precsim_2 s_2 \quad \text{and} \quad s_2 \precsim_{\frac{1}{2}} s_1.$$ ♦

If $s_1 \precsim_\varepsilon s_2$ and $\varepsilon \leq 1$, then this means that s_2 already simulates s_1, and ε gives a quantitative measure of how much s_2 is faster than s_1. On the other hand, if $\varepsilon > 1$, then s_2 does not simulate s_1, but if we accelerate the real-time behaviour of the entire process by ε, and consider s_2 in this accelerated process but s_1 still in the original process, then s_2 does simulate s_1. This is made precise by the following proposition.

Proposition 5.2.3. *For any $\varepsilon \in \mathbb{R}_{>0}$,*

$$s_1 \precsim_\varepsilon s_2 \quad \textit{if and only if} \quad s_1 \precsim (s_2)_\varepsilon$$

where $(s_2)_\varepsilon$ is a copy of s_2 where the entire process is accelerated by ε.

With the notion of ε-faster-than in hand, it is natural to ask what the smallest ε is such that the ε-faster-than relation holds, or in other words, what the smallest acceleration factor is that makes one process simulate another process. This motivates the definition of the simulation distance.

Definition 5.2.4. The *simulation distance* $d : S \times S \to [1, \infty]$ from a state s_1 to a state s_2 is given by

$$d(s_1, s_2) = \inf\{\varepsilon \geq 1 \mid s_1 \precsim_\varepsilon s_2\}.$$ ▲

For SMPs whose transition function is finitely supported, the simulation distance has the property that it is a generalisation of simulation in the sense that $s_1 \precsim s_2$ if and only if $d(s_1, s_2) = 1$.

The simulation distance itself is not a hemimetric, since it does not satisfy the triangle inequality. However, it satisfies a multiplicative version of the triangle inequality, which means that if we take the logarithm of the simulation distance, then we obtain a hemimetric.

Theorem 5.2.5. $\log d$ *is a hemimetric.*

5.3 Computing the Simulation Distance

Our first result on the simulation distance is that it can be computed, and the algorithm for computing it has polynomial complexity for the residence-time distributions we have considered so far. A key part of the algorithm is to consider, for two CDFs F and G, the value given by

$$c(F,G) = \inf\{\varepsilon \geq 1 \mid F \sqsubseteq_\varepsilon G\}.$$

Intuitively, $c(F,G)$ denotes the least acceleration factor required for F to be faster than G. Furthermore, given an SMP $\mathcal{M}$, let

$$\mathcal{C}(\mathcal{M}) = \{c(F_s, F_{s'}) \mid s, s' \in S\}.$$

We are interested in those SMPs for which we can actually compute $c(F,G)$ when F and G are residence-time distributions of the SMP.

Definition 5.3.1. An SMP $\mathcal{M}$ is *c-effective* if $\mathcal{C}(\mathcal{M})$ is computable. ▲

Given an SMP $\mathcal{M} = (S, \tau, \rho, \ell)$, we will denote by $f(l)$ the complexity of computing $c(F_s, F_{s'})$ for $s, s' \in S$, where l is the length of the representation of the residence-time distributions of $\mathcal{M}$.

Note that by Propositions 5.1.3-5.1.5, any SMP whose residence-time distributions are Dirac, uniform or exponential is c-effective and $f(l)$ is polynomial.

Lemma 5.3.2. *Let $\mathcal{M}$ be a finite SMP. If $d(s_1, s_2) \neq \infty$, then*

- *$s_1 \precsim_c s_2$, for some $c \in \mathcal{C}(\mathcal{M})$ and*
- *$d(s_1, s_2) = \min\{c \in \mathcal{C}(\mathcal{M}) \mid s_1 \precsim_c s_2\}$.*

Lemma 5.3.2 gives a strategy for computing $d(s_1, s_2)$. First we compute $\mathcal{C}(\mathcal{M})$, and then we check for each $c \in \mathcal{C}(\mathcal{M})$ whether $s_1 \precsim_c s_2$. If there is no such c, then $d(s_1, s_2) = \infty$, otherwise $d(s_1, s_2)$ will be the smallest c such that $s_1 \precsim_c s_2$. Hence we first need an algorithm to decide whether $s_1 \precsim_c s_2$. We do this by adapting to our setting the classic algorithm for deciding simulation for Markov chains [1, 5]. Given an SMP $\mathcal{M} = (S, \tau, \rho, \ell)$, let $n = |S|$ be the number of states, $m = |\mathtt{In}|$ the number of input actions, and $k = |\mathcal{AP}|$ the number of atomic propositions.

Theorem 5.3.3. *Let $\mathcal{M}$ be a finite and c-effective SMP. Given $s_1, s_2 \in S$ and $\varepsilon \geq 1$, deciding whether $s_1 \precsim_\varepsilon s_2$ can be done in time $\mathcal{O}(n^2(f(l)+k) + m^2 n)$.*

Using Theorem 5.3.3, we obtain the algorithm shown in Algorithm 5.3.1, which uses a bisection method to search through the elements of $\mathcal{C}(\mathcal{M})$ and test them, rather than simply testing them all.

Theorem 5.3.4. *Let $\mathcal{M}$ be a finite and c-effective SMP. The simulation distance between any two states can be computed in time $\mathcal{O}(n^2(f(l)+k) + m^2 n \cdot \log n)$.*

```
1  Order the elements of C(M) \ {∞} such that c_1 < c_2 < ··· < c_n ;
2  if s_1 ≾_{c_1} s_2 then return c_1 ;
3  else if s_1 ⋠_{c_n} s_2 then return ∞ ;
4  else
5  |  i ← 1, j ← n ;
6  |  while i < j do
7  |  |  h ← ⌈(j−i)/2⌉ ;
8  |  |  if s_1 ≾_{c_{j−h}} s_2 then j ← j − h ;
9  |  |  else i ← i + h ;
10 |  end
11 |  return c_j ;
12 end
```

Algorithm 5.3.1: Computing the simulation distance between two states s_1 and s_2.

5.4 Compositionality

The simulation distance turns out to behave nicely with respect to composition. More concretely, we prove that, under mild assumptions, composition is non-expansive with respective to the simulation distance. This result is a quantitative generalisation of the fact that simulation is a precongruence with respect to composition.

In order to obtain this result, we restrict our attention to those residence-time composition functions that are monotonic in the following sense.

Definition 5.4.1. A residence-time composition function $\star$ is *monotonic* if

$$F_\mu \sqsubseteq_\varepsilon F_\nu \quad \text{implies} \quad F_{\star(\mu,\eta)} \sqsubseteq_\varepsilon F_{\star(\nu,\eta)}$$

for all $\varepsilon \geq 1$ and $\mu, \nu, \eta \in \mathcal{D}(\mathbb{R}_{\geq 0})$. ▲

This is not a significant restriction, since most of the residence-time composition functions that are found in the literature are indeed monotonic. For monotonic residence-time composition functions we then have the promised non-expansiveness result.

Theorem 5.4.2. *For finite SMPs and monotonic $\star$,*

$$d(s_1, s_2) \leq \varepsilon \quad \textit{implies} \quad d(s_1 \parallel_\star s_3, s_2 \parallel_\star s_3) \leq \varepsilon.$$

The final aspect of compositionality that we will consider is how to compute the distance between composed states. We saw in Section 5.3 that we can compute the distance if we know how to compute the constants $c(F, G)$. For

a residence-time composition function on exponential distributions where we take e.g. the product of the rates, composition poses no problem, since the CDF of two exponential residence-time distributions composed in this manner is still an exponential distribution, and we know how to compute $c(F, G)$ for those.

However, when composing residence-time distributions using the pointwise maximum of their CDFs, the composition of two uniform distributions need not be a uniform distribution. Likewise, composing a uniform distribution and an exponential distribution in this manner yields a CDF that is neither uniform nor exponential. Hence we must consider how to compute $c(F, G)$ for these composed distributions.

Proposition 5.4.3. *Let $\star$ be maximum composition. The constants $c(F_\mu, F_{\star(\nu,\eta)})$ and $c(F_{\star(\mu,\eta)}, F_\nu)$ are computable whenever μ, ν, and η are taken from the set of exponential, uniform, and Dirac distributions.*

The proof of Proposition 5.4.3 is laborious and tedious, because many combinations and special cases must be considered. Hence we have not extended the result to a higher number of compositions or to composition on both sides, although we strongly believe that such a result will hold for many other kinds of distributions also.

5.5 Logical Properties

If the simulation distance tells us that two processes are close, then we would also expect them to satisfy almost the same properties. In this section we make this idea precise by introducing a logical specification language for specifying properties of SMPs. We show that this language characterises ε-simulation and that the simulation distance from s_1 to s_2 is less than ε if and only if whenever s_1 satisfies a formula, s_2 satisfies a slight perturbation of the same formula.

The language we use is a slight extension of Markovian logic [2] which we call *timed Markovian logic* (TML). TML has the following syntax, where $\alpha \in \mathcal{AP}$, $a \in \mathtt{In}$, $p \in \mathbb{Q} \cap [0, 1]$, and $t \in \mathbb{Q}_{\geq 0}$.

$$\text{TML}: \quad \varphi ::= \alpha \mid \neg\alpha \mid \ell_p t \mid m_p t \mid L^a_p \varphi \mid M^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

α and $\neg\alpha$ speak about atomic propositions, $\wedge$ and $\vee$ are the usual conjunction and disjunction, $\ell_p t$ and $m_p t$ speak about the timing behaviour of processes, and L^a_p and M^a_p speak about the branching behaviour of processes. This is made precise by the semantics of TML, which are given as follows.

$$
\begin{array}{llll}
s \models \alpha & \text{iff} & \alpha \in \ell(s) & \qquad s \models \ell_p t \quad \text{iff} \quad F_s(t) \geq p \\
s \models \neg\alpha & \text{iff} & \alpha \notin \ell(s) & \qquad s \models m_p t \quad \text{iff} \quad F_s(t) \leq p \\
s \models \varphi \wedge \varphi' & \text{iff} & s \models \varphi \text{ and } s \models \varphi' & \qquad s \models L^a_p \varphi \quad \text{iff} \quad \tau(s,a)([\![\varphi]\!]) \geq p \\
s \models \varphi \vee \varphi' & \text{iff} & s \models \varphi \text{ or } s \models \varphi' & \qquad s \models M^a_p \varphi \quad \text{iff} \quad \tau(s,a)([\![\varphi]\!]) \leq p
\end{array}
$$

where $[\![\varphi]\!]$ is the set of states satisfying φ.

Furthermore, we will consider the following fragments of TML.

$$\text{TML}^{\geq}: \quad \varphi ::= \alpha \mid \neg\alpha \mid \ell_p t \mid L^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

$$\text{TML}^{\leq}: \quad \varphi ::= \alpha \mid \neg\alpha \mid m_p t \mid M^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

In order to connect TML to our simulation distance, we introduce the notion of ε-*perturbance* of a formula $\varphi \in \text{TML}$, denoted by $(\varphi)_\varepsilon$, which is given by replacing all occurrences of $\ell_p t$ or $m_p t$ in φ by $\ell_p \varepsilon \cdot t$ or $m_p \varepsilon \cdot t$, respectively. We then get the following result, which shows that the fragments $\text{TML}^{\leq}$ and $\text{TML}^{\geq}$ characterise ε-simulation.

Theorem 5.5.1. *Let $\varepsilon \in \mathbb{Q}_{\geq 0}$ with $\varepsilon \geq 1$. Then the following holds.*

- $s_1 \precsim_\varepsilon s_2$ *if and only if* $\forall \varphi \in \text{TML}^{\geq}. s_1 \models \varphi \implies s_2 \models (\varphi)_\varepsilon$.
- $s_1 \precsim_\varepsilon s_2$ *if and only if* $\forall \varphi \in \text{TML}^{\leq}. s_2 \models (\varphi)_\varepsilon \implies s_1 \models \varphi$.

From this result we get the following corollary, which connects the fragments of TML directly to our simulation distance.

Corollary 5.5.2. *Let $\varepsilon \in \mathbb{Q}_{\geq 0}$ with $\varepsilon \geq 1$. For finite SMPs the following holds.*

- $d(s_1, s_2) \leq \varepsilon$ *if and only if* $\forall \varphi \in \text{TML}^{\geq}. s_1 \models \varphi \implies s_2 \models (\varphi)_\varepsilon$.
- $d(s_1, s_2) \leq \varepsilon$ *if and only if* $\forall \varphi \in \text{TML}^{\leq}. s_2 \models (\varphi)_\varepsilon \implies s_1 \models \varphi$.

Another property that is often of interest is that of *reachability*: Starting from a given state, can we reach a state which satisfies some property? For SMPs which have both probabilistic branching and real-time behaviour, a more interesting reachability problem is:

> Starting from a given state s, can we with probability at least p and before time t reach a state which satisfies property φ?

We will now show that the ε-simulation relation also preserves reachability properties. In order to do this, we consider the following kind of events. Let $X \subseteq S$ and $t \in \mathbb{R}_{\geq 0}$. Then we define

$$\Diamond^t X = \{\pi \in \Pi(M) \mid \exists i \in \mathbb{N}. \pi[i] \in X \text{ and } \sum_{j=1}^{i-1} \pi\langle j \rangle \leq t\}$$

as the set of paths that eventually reach a state in X within time t. We can then prove the following theorem.

Theorem 5.5.3. *Let β be a Boolean combination of atomic propositions. If we have $s_1 \precsim_\varepsilon s_2$, then for any scheduler σ there exists a scheduler σ' such that*

$$\mathbb{P}^{\sigma}_{s_1}(\Diamond^{t}[\![\beta]\!]) \leq \mathbb{P}^{\sigma'}_{s_2}(\Diamond^{\varepsilon \cdot t}[\![\beta]\!]).$$

In other words, $s_1 \precsim_\varepsilon s_2$ guarantees that whenever s_1 can reach a state in $[\![\beta]\!]$ within time t under some scheduler σ, then we can find a scheduler σ' such that s_2 reaches a state in $[\![\beta]\!]$ within time $\varepsilon \cdot t$, and with at least as high probability.

5.6 Topology of the Simulation Distance

In this section we investigate the topology of the simulation distance, in particular with respect to properties expressible in TML. The question we wish to answer is whether, given a sequence of states $\{s_k\}$ that converges to a state s such that $s_i \models \varphi$ for each state s_i in the sequence, we can be sure that also the state s satisfies φ. This is the same as asking whether the set $[\![\varphi]\!]$ is a closed set in the topology induced by the simulation distance. Hence, if $[\![\varphi]\!]$ is closed, reasoning in the limit about properties expressible by φ is sound.

Recall that the *right-centered* topology is generated by the open balls

$$\mathcal{B}^R_r(s) = \{s' \mid d(s', s) < r\}$$

and the *left-centered* topology is generated by the open balls

$$\mathcal{B}^L_r(s) = \{s' \mid d(s, s') < r\}.$$

Lemma 5.6.1. *The following holds in the right-centered topology.*

1. *$[\![\ell_p t]\!]$ is closed.*
2. *If $p = 0$, then $[\![\ell_p t]\!]$ is open.*
3. *If $p > 0$, then $[\![\ell_p t]\!]$ is not open.*
4. *If $p = 1$, then $[\![m_p t]\!]$ is closed.*
5. *If $p < 1$, then $[\![m_p t]\!]$ is not closed.*

We can then use Lemma 5.6.1 to show that reasoning in the limit is sound for the right-centered topology.

Theorem 5.6.2. *For any $\varphi \in \mathrm{TML}^{\geq}$, $[\![\varphi]\!]$ is closed in the right-centered topology.*

Lemma 5.6.3. *The following holds in the left-centered topology.*

1. *If $p = 1$, then $[\![m_p t]\!]$ is open.*
2. *If $p < 1$, then $[\![m_p t]\!]$ is not open.*
3. *If $p = 0$, $[\![\ell_p t]\!]$ is closed.*
4. *If $p > 0$, $[\![\ell_p t]\!]$ is not closed.*

However, we have not been able to determine whether $[\![m_p t]\!]$ is closed in the left-centered topology or not. If it were closed, then we could prove that reasoning in the limit is also sound for the left-centered topology. Our intuition leads us to conjecture that this is the case.

Conjecture 5.6.4. *For any $\varphi \in \mathrm{TML}^{\leq}$, $[\![\varphi]\!]$ is closed in the left-centered topology.*

5.7 References

[1] C. Baier, B. Engelen, and M. E. Majster-Cederbaum, "Deciding bisimilarity and similarity for probabilistic processes," *J. Comput. Syst. Sci.*, vol. 60, no. 1, pp. 187–231, 2000. [Online]. Available: http://dx.doi.org/10.1006/jcss.1999.1683

[2] D. Kozen, R. Mardare, and P. Panangaden, "Strong completeness for Markovian logics," in *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and J. Sgall, Eds., vol. 8087. Springer, 2013, pp. 655–666. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40313-2_58

[3] M. R. Pedersen, G. Bacci, K. G. Larsen, and R. Mardare, "A hemimetric extension of simulation for semi-Markov decision processes," in *Quantitative Evaluation of Systems - 15th International Conference, QEST 2018, Beijing, China, September 4-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science, A. McIver and A. Horvath, Eds., vol. 11024. Springer, 2018, pp. 339–355. [Online]. Available: https://doi.org/10.1007/978-3-319-99154-2_21

[4] M. Shaked and G. Shanthikumar, *Stochastic Orders*, ser. Springer Series in Statistics. Springer, 2007.

[5] L. Zhang, "Decision algorithms for probabilistic simulations," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2009. [Online]. Available: http://scidok.sulb.uni-saarland.de/volltexte/2009/2424/

Chapter 6

Conclusion

In this thesis we have investigated how to reason about the real-time behaviour of stochastic systems, both by expressing and verifying properties in a given specification language, and by comparing systems through different notions of behavioural relations. Through this we have developed various logical specification languages as well as new ways of comparing systems, and studied the properties of these.

We have introduced weighted logic with bounds (WLWB) as a specification language, which allows one to reason about upper and lower bounds on weights in a weighted transition system. Since many requirements of interest speak about upper and lower bounds, we argue that this is a useful language for specifying requirements. Furthermore, this language is less susceptible to the approximate modelling problem, since it does not speak about exact weights. We have studied the properties of this language and shown that it characterises a kind of behavioural equivalence that looks at the upper and lower bounds of transitions rather than matching each weight exactly. We have also given a complete axiomatisation of the language, and developed algorithms for deciding both the model checking and the satisfiability problem for WLWB.

We have defined a notion of a faster-than relation on traces for semi-Markov processes by requiring that on each trace, the fast process must have a higher probability of completing that trace within any given time bound than the slow process. Such a notion is useful, since it allows one to reason about incremental improvement of a system in the design phase, where the speed of the system can be gradually improved. It is also useful from a compositional perspective. From this perspective, one may identify a component that is working too slowly, and replace it with a component that is faster. However, in this case one has to be careful about timing anomalies. We have shown that such timing anomalies can occur, and we have taken first steps

toward avoiding them, by identifying conditions that are sufficient for guaranteeing the absence of timing anomalies. We have also shown that deciding the faster-than relation is a difficult problem that is undecidable in general. Moreover, it can not be approximated up to a multiplicative constant, due to a close connection to probabilistic automata. Despite this, we have given an algorithm for approximating a time-bounded variant of the faster-than problem up to an additive constant, as well as an algorithm for deciding the faster-than relation exactly when restricting to unambiguous processes.

Since the qualitative answers of the classical notion of simulation for semi-Markov processes are too rigid for the quantitative nature of these processes, we have extended the concept of simulation to a quantitative simulation distance. This distance gives information about how much one should increase the speed of a process in order for it to become as fast as another process. This is useful when you have a system which does not satisfy a given property, but you have a model of a system which does. Then the distance tells you how much you need to increase the speed of the original system in order for it to also satisfy the property. We have shown how to efficiently compute this distance for some commonly used residence-time distributions, and that composition is non-expansive with respect to the distance, meaning that timing anomalies can not occur. Furthermore, we have shown that the distance can be characterised by a logical specification language which we call timed Markovian logic, and that it preserves reachability properties. We have also shown that, in the topology induced by the simulation distance, properties expressed in timed Markovian logic are preserved in the limit, in the sense that if a sequence of states has converges to a certain state, and if each state in the sequence satisfies some property expressed in timed Markovian logic, then the state to which the sequence converges will also satisfy that property.

The research presented here has been carried out as part of the research project Approximate Reasoning for Stochastic Markovian Systems, which is funded by The Danish Council for Independent Research. The content of this thesis contributes to the project by introducing formalisms and algorithms for approximating the behaviour of stochastic system, and studying the properties of these from a logical, topological, as well as computational point of view.

6.1 Future Work

The work presented here has contributed to our understanding of how to reason about and compare the stochastic behaviour of systems from a logical, topological, as well as computational point of view. However, there still remain many open problems that we intend to investigate in future work. We discuss here the most important of these.

Strong completeness and Stone duality. The completeness result we have shown for WLWB is a weak completeness result, showing that

$$\models \varphi \quad \text{implies} \quad \vdash \varphi$$

for any formula φ in WLWB. The notion of strong completeness asks that

$$\Phi \models \varphi \quad \text{implies} \quad \Phi \vdash \varphi$$

for any formula φ and set of formulas Φ in WLWB. Proving strong completeness would require additional, infinitary axioms such as the rules

$$\{L_q\varphi \mid q < r\} \vdash L_r\varphi \quad \text{and} \quad \{M_q\varphi \mid q < r\} \vdash M_r\varphi$$

for a given r, which describe the Archimedean property of the reals.

A strong completeness result may also point the way to a Stone duality result [9, 10, 17], which sheds light on the connection between logic and topology. Such results have already been developed for Markovian logics [8, 12, 13], which bear some similarity to WLWB.

Extend logical specification languages with temporal operators. The logical specification languages we have introduced and studied are all quite parsimonious, although our results show that they are expressive enough to characterise the relations under consideration. However, when specifying requirements in an actual engineering situation, it may be useful to introduce additional constructs in the language in order to allow for more expressivity. In particular, it would be interesting to add temporal operators like those found in LTL [15] and CTL [4], such as "until" and "eventually", or even fixpoint operators like those found in the μ-calculus [11], and investigate how many of our results carry over to this more expressive language.

Better understanding of timing anomalies. Although we shown that timing anomalies can occur when reasoning with the faster-than relation, and we have given conditions that are sufficient to guarantee that no timing anomalies occur, a complete understanding of when and how timing anomalies occur is still missing. First of all, the conditions that we have given are very restrictive, and requires that the processes in question are fully deterministic. Secondly, the conditions impose requirements on all the involved processes, and not just the context. It would be preferable to have conditions that only look at the context, since then one could verify the context once and for all, and then be guaranteed that no timing anomalies occur when swapping components in and out. One way to try and gain a better understanding of timing anomalies may be to consider networks of priced timed automata [7] instead of semi-Markov processes, since the former are more well-suited for compositional reasoning.

Another aspect that is missing from our understanding of timing anomalies is what happens in the generative case. We have only considered reactive processes here, since defining composition for these is more natural than for generative processes [16]. However, several notions of composition have been defined for generative processes [2, 5, 6, 18], and we are interested in seeing how timing anomalies behave and can be avoided in this setting.

Develop algorithms for deciding the faster-than relation for reactive systems. For both reactive and generative processes, we have shown that the faster-than relation is undecidable in general. However, for generative processes, we have nonetheless been able to develop algorithms for two special cases, namely those of time-bounded additive approximation and unambiguous processes. These cases do not immediately carry over to the setting of reactive systems, where the main challenge is that of handling the schedulers involved, of which there may be uncountably many, depending on the kind of scheduler under consideration [19]. We have extended the result on time-bounded additive approximation from generative processes to reactive processes, but only for the case where there are countably many schedulers, meaning that the schedulers may not take time into account. It is therefore still unclear to us whether there exists an algorithm for the case of uncountably many schedulers. The same comments also apply to the algorithm for unambiguous processes.

Take probabilistic branching into account in the simulation distance. One weakness of the simulation distance is that it only considers differences in the residence-time distributions and not the differences in the probabilistic branching. This is because we have chosen to focus on the real-time behaviour of systems. However, there are some cases where this is not completely satisfactory.

Consider for example the semi-Markov process depicted in Figure 6.1.1. In this case, we have $s_1 \precsim t_1$ whenever $\theta \leq 5$. However, if θ is perturbed just a little bit above 5, then we get $s_1 \not\precsim t_1$. For example, consider what happens when $\theta = 5.00001$. Then it is not difficult to see that the simulation distance will be

$$d(s_1, t_1) = \frac{5.00001}{5} = 1.000002.$$

However, for so small perturbations of θ, one may argue in the following way that t_1 should be considered to be faster than s_1. While it is true that t_2 is slower than s_2, and that t_1 has a non-zero probability of transitioning to t_2, t_2 is only marginally slower than s_2, and furthermore, t_1 transitions only to t_2 10% of the time. The remaining 90% of the time, t_1 will transition to t_3, which is significantly faster than s_2. Hence, the greater probability of going to a much faster state should somehow outweigh the small probability of

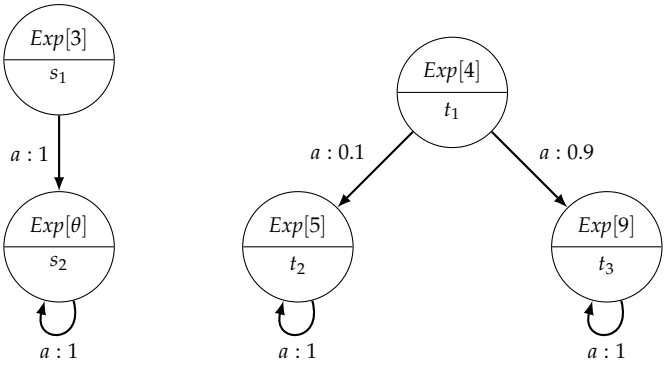


Figure 6.1.1: A semi-Markov process where $s_1 \precsim t_1$ if $\theta \leq 5$ and $s_1 \not\precsim t_1$ if $\theta > 5$.

going to a slightly slower state.

We believe that the kind of reasoning just described will be possible by taking into consideration also the difference between the probabilistic branching of the processes in defining the simulation distance. For this, an obvious possibility would be to incorporate the (non-symmetric) Kantorovich distance (see e.g. [3]). This is somewhat similar to what has been done for bisimulation distances on continuous-time Markov processes by combining the Kantorovich distance and the total variation distance [1]. However, it is not clear to us at present how the distance between the real-time behaviour in the states and the distance between the transition distributions should interact.

Consider behavioural distances starting from the topological point of view. In this thesis and in other works such as [13, 14], topological issues of simulation and bisimulation distances have been investigated, in particular how properties of the system behave with respect to the topology. However, in these cases, the distance comes before the topology in the sense that the distance is defined and the topology is then investigated. We believe that in order to understand better the interplay between (bi)simulation, distances, topology, and logical properties, it will be beneficial to define first the topology that characterises (bi)simulation, and then see what distances can metrise this topology.

6.2 Summary

The research presented in this thesis makes a novel contribution to challenging problems encountered when dealing with stochastic systems. This research has deepened our understanding of how to compare and express properties about stochastic systems, as well as opened new lines of research

that the author intends to pursue in the future.

6.3 References

[1] G. Bacci, G. Bacci, K. G. Larsen, and R. Mardare, "On-the-fly computation of bisimilarity distances," *Logical Methods in Computer Science*, vol. 13, no. 2, 2017. [Online]. Available: https://doi.org/10.23638/LMCS-13(2:13)2017

[2] J. C. M. Baeten, J. A. Bergstra, and S. A. Smolka, "Axiomatizing probabilistic processes: ACP with generative probabilities," *Inf. Comput.*, vol. 121, no. 2, pp. 234–255, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1135

[3] K. Chatzikokolakis, "On the additive capacity problem for quantitative information flow," in *Quantitative Evaluation of Systems - 15th International Conference, QEST 2018, Beijing, China, September 4-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science, A. McIver and A. Horvath, Eds., vol. 11024. Springer, 2018, pp. 1–19. [Online]. Available: https://doi.org/10.1007/978-3-319-99154-2_1

[4] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching-time temporal logic," in *Logics of Programs, Workshop, Yorktown Heights, New York, May 1981*, ser. Lecture Notes in Computer Science, D. Kozen, Ed., vol. 131. Springer, 1981, pp. 52–71. [Online]. Available: https://doi.org/10.1007/BFb0025774

[5] R. Cleaveland, S. A. Smolka, and A. E. Zwarico, "Testing preorders for probabilistic processes," in *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, ser. Lecture Notes in Computer Science, W. Kuich, Ed., vol. 623. Springer, 1992, pp. 708–719. [Online]. Available: https://doi.org/10.1007/3-540-55719-9_116

[6] P. R. D'Argenio, H. Hermanns, and J. Katoen, "On generative parallel composition," *Electr. Notes Theor. Comput. Sci.*, vol. 22, pp. 30–54, 1999. [Online]. Available: https://doi.org/10.1016/S1571-0661(05)80596-1

[7] A. David, K. G. Larsen, A. Legay, M. Mikucionis, D. B. Poulsen, J. van Vliet, and Z. Wang, "Statistical model checking for networks of priced timed automata," in *Formal Modeling and Analysis of Timed Systems - 9th International Conference, FORMATS 2011, Aalborg, Denmark, September 21-23, 2011. Proceedings*, ser. Lecture Notes in Computer Science,

U. Fahrenberg and S. Tripakis, Eds., vol. 6919. Springer, 2011, pp. 80–96. [Online]. Available: https://doi.org/10.1007/978-3-642-24310-3_7

[8] R. Furber, D. Kozen, K. G. Larsen, R. Mardare, and P. Panangaden, "Unrestricted stone duality for Markov processes," in *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*. IEEE Computer Society, 2017, pp. 1–9. [Online]. Available: https://doi.org/10.1109/LICS.2017.8005152

[9] S. Givant and P. Halmos, *Introduction to Boolean Algebras*, ser. Undergraduate Texts in Mathematics. Springer, 2009.

[10] P. T. Johnstone, *Stone Spaces*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1986.

[11] D. Kozen, "Results on the propositional mu-calculus," *Theor. Comput. Sci.*, vol. 27, pp. 333–354, 1983. [Online]. Available: https://doi.org/10.1016/0304-3975(82)90125-6

[12] D. Kozen, K. G. Larsen, R. Mardare, and P. Panangaden, "Stone duality for Markov processes," in *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 2013, pp. 321–330. [Online]. Available: http://dx.doi.org/10.1109/LICS.2013.38

[13] D. Kozen, R. Mardare, and P. Panangaden, "A metrized duality theorem for Markov processes," *Electr. Notes Theor. Comput. Sci.*, vol. 308, pp. 211–227, 2014. [Online]. Available: https://doi.org/10.1016/j.entcs.2014.10.012

[14] K. G. Larsen, R. Mardare, and P. Panangaden, "Taking it to the limit: Approximate reasoning for Markov processes," in *Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, ser. Lecture Notes in Computer Science, B. Rovan, V. Sassone, and P. Widmayer, Eds., vol. 7464. Springer, 2012, pp. 681–692. [Online]. Available: https://doi.org/10.1007/978-3-642-32589-2_59

[15] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*. IEEE Computer Society, 1977, pp. 46–57. [Online]. Available: https://doi.org/10.1109/SFCS.1977.32

[16] A. Sokolova and E. P. de Vink, "Probabilistic automata: System types, parallel composition and comparison," in *Validation of Stochastic Systems - A Guide to Current Research*, ser. Lecture Notes in Computer

Science, C. Baier, B. R. Haverkort, H. Hermanns, J. Katoen, and M. Siegle, Eds., vol. 2925. Springer, 2004, pp. 1–43. [Online]. Available: https://doi.org/10.1007/978-3-540-24611-4_1

[17] M. H. Stone, "The theory of representation for Boolean algebras," *Transactions of the American Mathematical Society*, vol. 40, no. 1, pp. 37–111, 1936. [Online]. Available: http://www.jstor.org/stable/1989664

[18] R. J. van Glabbeek, S. A. Smolka, and B. Steffen, "Reactive, generative and stratified models of probabilistic processes," *Inf. Comput.*, vol. 121, no. 1, pp. 59–80, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1123

[19] N. Wolovick and S. Johr, "A characterization of meaningful schedulers for continuous-time Markov decision processes," in *Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, September 25-27, 2006, Proceedings*, ser. Lecture Notes in Computer Science, E. Asarin and P. Bouyer, Eds., vol. 4202. Springer, 2006, pp. 352–367. [Online]. Available: https://doi.org/10.1007/11867340_25

Part II

Papers

Paper A

Reasoning About Bounds in Weighted Transition Systems

Mikkel Hansen, Kim Guldstrand Larsen, Radu Mardare, and Mathias Ruggaard Pedersen

The paper is under submission to Logical Methods in Computer Science.
The paper is an extended version of [16].

Abstract

We propose a way of reasoning about minimal and maximal values of the weights of transitions in a weighted transition system (WTS). This perspective induces a notion of bisimulation that is coarser than the classic bisimulation: it relates states that exhibit transitions to bisimulation classes with the weights within the same boundaries. We propose a customised modal logic that expresses these numeric boundaries for transition weights by means of particular modalities. We prove that our logic is invariant under the proposed notion of bisimulation. We show that the logic enjoys the finite model property and we identify a complete axiomatisation for the logic. Last but not least, we use a tableau method to show that the satisfiability problem for the logic is decidable.

A.1 Introduction

Weighted transition systems (WTSs) are used to model concurrent and distributed systems in the case where some resources are involved, such as time, bandwidth, fuel, or energy consumption. Recently, the concept of a cyber-physical system (CPS), which considers the integration of computation and the physical world has become relevant in modeling various real-life situations. In these models, sensor feedback affects computation, and through machinery, computation can further affect physical processes. The quantitative nature of weighted transition systems is well-suited for the quantifiable inputs and sensor measurements of CPSs, but their rigidity makes them less well-suited for the uncertainty inherent in CPSs. In practice, there is often some uncertainty attached to the resource cost, whereas weights in a WTS are precise. Thus, the model may be too restrictive and unable to capture the uncertainties inherent in the domain that is being modeled.

In this paper, we attempt to remedy this shortcoming by introducing a modal logic for WTSs that allows for approximate reasoning by speaking about upper and lower bounds for the weights of the transitions. The logic has two types of modal operators that reason about the minimal and maximal weights on transitions, respectively. This allows reasoning about models where the quantitative information may be imprecise (e.g. due to imprecisions introduced when gathering real data), but where we can establish a lower and upper bound for transitions.

In order to provide the semantics for this logic, we use the set of possible transition weights from one state to a set of states as an abstraction of the actual transition weights. The logic is expressive enough to characterise WTSs up to a relaxed notion of weighted bisimilarity, where the classical conditions are replaced with conditions requiring that the minimal and maximal weights on transitions are matched.

Our main contribution is a complete axiomatisation of our logic, showing that any validity in this logic can be proved as a theorem from the axiomatic system. Completeness allows us to transform any validity checking problem into a theorem proving one that can be solved automatically by modern theorem provers, thus bridging the gap to the theorem proving community. The completeness proof adapts the classical filtration method, which allows one to construct a (canonical) model using maximal consistent sets of formulas. The main difficulty of adapting this method to our setting is that we must establish both lower and upper bounds for the transitions in this model. To achieve this result, we demonstrate that our logic enjoys the finite model property.

Our second significant contribution is a decision procedure for determining the satisfiability of formulas in our logic. This decision procedure makes use of the tableau method to construct a tableau for a given formula. If the constructed tableau is successful, then the formula is satisfiable, and a finite model for the formula can be generated from the tableau.

Related Work.

In [12], Zoltán Ésik also considered the issue of bisimulation for weighted transition systems, although in the more general setting of synchronisation trees with weights in an arbitrary monoid or semiring. Synchronisation trees arise by unfolding the transitions of a weighted transition system starting in some state which will become the root of the tree. Both Ésik's and our notion of bisimilarity bears some resemblance to probabilistic bisimulation [29], by considering not only single transitions but transitions to equivalence classes of states. However, while we require that the upper and lower bounds of these transitions should match, the bisimilarity of Ésik requires that the sum of the transitions should be the same. This is motivated by the fact that the synchronisation trees do not form a category which respects the additive structure of a semiring. However, as Ésik proves, if one takes the quotient with respect to his version of weighted bisimilarity, then the category one obtains does respect the additive structure. Thus, the semiring structure of the weights is of vital importance to Ésik's work, but is an aspect that we have not considered in our work.

Several logics have been proposed in the past to express properties of quantified (weighted, probabilistic or stochastic) systems. They typically use modalities indexed with real numbers to express properties such as *"φ holds with at least probability b"*, *"we can reach a state satisfying φ with a cost at least r"*, etc.

In the context of weighted automata, weighted monadic second order logic has been introduced by Droste and Gastin [9] to capture the behaviour of weighted automata for commutative semirings. This work has been ex-

tended to many closely related systems [2, 10, 11, 14, 31]. There has also been work on connecting weighted monadic second order logic with probabilistic CTL [4]. For weighted transition systems, weighted modal logic has been introduced by Larsen and Mardare [23] to reason about the consumption of resources in such a system. This logic has been extended to handle recursion [27, 28] as well as parallel composition and concurrency [25]. For both the original weighted modal logic and its concurrent extension, complete axiomatisations were developed. A weighted extension of the μ-calculus was introduced by Larsen et al. in [24], where a complete axiomatisation for this extension was also given.

While our setting is that of weighted transition systems, our logic and the development of its theory has more in common with Markovian logic than with the previously mentioned work on weighted systems. Markovian logic was introduced by Mardare et al. [5, 30] building on previous work on probability logics [13, 17, 32]. Markovian logic reasons about probabilistic and stochastic systems using operators L_r and M_r which mean that a property hold with *at least* probability r or *at most* probability r, respectively. Much of the work on Markovian logic has focused on giving a complete axiomatisation for the logic [22], culminating in a Stone duality for Markov processes [21]. However, compositional aspects have been considered in [6], where also an axiomatisation was given for Markovian logic with an operator for parallel composition.

While our logical syntax resembles that of Markovian logic, our semantics is different in the sense that we argue not about probabilities, but about an interval of possible weights. For instance, in the aforementioned logics we have a validity of type $\vdash \neg L_r\phi \to M_r\phi$ saying that the value of the transition from the current state to ϕ is either at least r or at most r; on the other hand, in our logic the formula $\neg L_r\phi \wedge \neg M_r\phi$ might have a model since $L_r\phi$ and $M_r\phi$ express the fact that the lower cost of a transition to ϕ is at least r and the highest cost is at most r respectively.

Our completeness proof uses a technique similar to the one used for weighted modal logic [23] and Markovian logic [5, 22, 30]. It is however different from these related constructions since our axiomatisation is finitary, while the aforementioned ones require infinitary proof rules. Our axiomatic systems are related to the ones mentioned above and the mathematical structures revealed by this work are also similar to the related ones. This suggest a natural extension towards a Stone duality result along the lines of [21], which we will consider in a future work.

Decidability results regarding satisfiability have also been given for some related logics, such as weighted modal logic [26] and probabilistic versions of CTL and the μ-calculus [7]. However, the satisfiability problem is known to be undecidable for other related logics, in particular timed logics such as TCTL [1] and timed modal logic [18]. This fact suggests that our logic is an

interesting one which, despite its expressivity, remains decidable.

Our approach of considering upper and lower bounds is related to work on interval-based formalisms such as interval Markov chains (IMCs) [19] and interval weighted modal transition systems (WMTSs) [20]. Much like our approach, IMCs consider upper and lower bounds on transitions in the probabilistic case. WMTSs add intervals of weights to individual transitions of modal transition systems, in which there can be both may- and must-transitions. A main focus of the work both on IMCs and WMTSs have been a process of refinement, making the intervals progressively smaller until an implementation is obtained. However, none of these works have explored the logical perspective up to the level of axiomatisation or satisfiability results, which is the focus of our paper.

A.2 Model

The models addressed in this paper are weighted transition systems, in which transitions are labelled with numbers to specify the cost of the corresponding transition. In order to specify and reason about properties regarding imprecision, such as "the maximum cost of going to a safe state is 10" and "the minimum cost of going to a halting state is 5", we will abstract away the individual transitions and only consider the minimum and maximum costs from a state to another. We will do this by constructing for any two states the set of weights that are allowed from one to the other.

First we recap the definition of a weighted transition system. Let $\mathcal{AP}$ be a countable set of atomic propositions. A WTS is formally defined as follows:

Definition A.2.1. A *weighted transition system (WTS)* is a tuple $\mathcal{M} = (S, \to, \ell)$, where

- S is a non-empty set of *states*,
- $\to \subseteq S \times \mathbb{R}_{\geq 0} \times S$ is the *transition relation*, and
- $\ell : S \to 2^{\mathcal{AP}}$ is a *labelling function* mapping to each state a set of atomic propositions. ▲

Note that we impose no restrictions on the state space S; it can be uncountable. We write $s \xrightarrow{r} t$ to mean that $(s, r, t) \in \to$. We will say that a WTS is *image-finite* if for any $s \in S$ there are only finitely many $t \in S$ such that $s \xrightarrow{r} t$ for some $r \in \mathbb{R}_{\geq 0}$.

When modeling cyber-physical systems, it is often unreasonable to expect one to know the exact weights for transitions. However, it is often the case that one has some bounds on the actual weights, e.g. one might know that the cost of taking some transition is between 5 and 25. In order to reason

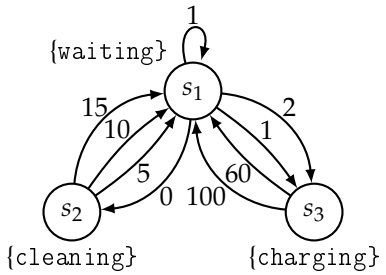


Figure A.2.1: A simple model of a robot vacuum cleaner.

about these bounds, we abstract away the individual transitions, and instead consider the set of weights between a state and a set of states.

Definition A.2.2. For an arbitrary WTS $\mathcal{M} = (S, \to, \ell)$, the function $\theta_{\mathcal{M}} : S \to \left(2^S \to 2^{\mathbb{R}_{\geq 0}}\right)$ is defined for any state $s \in S$ and set of states $T \subseteq S$ as

$$\theta_{\mathcal{M}}(s)(T) = \{r \in \mathbb{R}_{\geq 0} \mid \exists t \in T \text{ such that } s \xrightarrow{r} t\}. \qquad \blacktriangle$$

Thus $\theta_{\mathcal{M}}(s)(T)$ is the set of all possible weights of going from s to a state in T. We will sometimes refer to $\theta(s)(T)$ as the *image from s to T* or simply as an *image set*. In the rest of the paper, we will use the notation

$$\theta^-(s)(T) = \begin{cases} -\infty & \text{if } \theta(s)(T) = \emptyset \\ \inf \theta(s)(T) & \text{otherwise} \end{cases}$$

and

$$\theta^+(s)(T) = \begin{cases} \infty & \text{if } \theta(s)(T) = \emptyset \\ \sup \theta(s)(T) & \text{otherwise.} \end{cases}$$

Thus $\theta^-(s)(T)$ is a lower bound on the weights from s to T and $\theta^+(s)(T)$ is an upper bound.

Example A.2.3. Figure A.2.1 shows a simple model of a robot vacuum cleaner that can be in a waiting state, a cleaning state, or a charging state. This is an example of a cyber-physical system where the costs of transitions are necessarily imprecise. The time it takes to recharge the batteries depends on the condition of the batteries as well as that of the charger; the time it takes to clean the room depends on how dirty the room is, and how free the floor is from obstacles; and the time it takes to reach the charger depends on where in the room the robot is when it needs to be recharged. By constructing the image sets, we can abstract away from the individual transitions. For example, we have $\theta(s_2)(\{s_1\}) = \{5, 10, 15\}$, so $\theta^-(s_2)(\{s_1\}) = 5$ and $\theta^+(s_2)(\{s_1\}) = 15$. ♦

We will now establish some useful properties of image sets. In particular, the transition function is monotonic with respect to set inclusion, and union distributes over image sets as one might expect.

Lemma A.2.4 (Monotonicity of θ)**.** *Let $\mathcal{M} = (S, \to, \ell)$ be a WTS and let T_1 and T_2 be subsets of S. If $T_1 \subseteq T_2$, then $\theta(s)(T_1) \subseteq \theta(s)(T_2)$.*

Lemma A.2.5. *Let $\mathcal{M} = (S, \to, \ell)$ be a WTS. For any $s \in S$ and $T_1, T_2 \subseteq S$, it holds that*

1. $\theta(s)(T_1 \cup T_2) = \theta(s)(T_1) \cup \theta(s)(T_2)$ *and*
2. $\theta(s)(T_1 \cap T_2) \subseteq \theta(s)(T_1) \cap \theta(s)(T_2)$.

As usual we would like some way of relating model states with equivalent behavior. To this end we define the notion of a bisimulation relation. The classical notion of a bisimulation relation for weighted transition systems [3], which we term weighted bisimulation, is defined as follows.

Definition A.2.6. Given a WTS $\mathcal{M} = (S, \to, \ell)$, an equivalence relation $\mathcal{R} \subseteq S \times S$ on S is called a *weighted bisimulation relation* if and only if for all $s, t \in S$, $s\mathcal{R}t$ implies

- (Atomic harmony) $\ell(s) = \ell(t)$,
- (Zig) if $s \xrightarrow{r} s'$ then there exists $t' \in S$ such that $t \xrightarrow{r} t'$ and $s'\mathcal{R}t'$, and
- (Zag) if $t \xrightarrow{r} t'$ then there exists $s' \in S$ such that $s \xrightarrow{r} s'$ and $s'\mathcal{R}t'$. ▲

We say that $s, t \in S$ are weighted bisimilar, written $s \sim_W t$, if and only if there exists a weighted bisimulation relation $\mathcal{R}$ such that $s\mathcal{R}t$. Weighted bisimilarity, $\sim_W$, is the largest weighted bisimulation relation.

Since it is our goal to abstract away from the exact weights on the transitions, the bisimulation that we will now introduce does not impose the classical zig-zag conditions [3] of a bisimulation relation, but instead require that bounds be matched for any bisimulation class.

Definition A.2.7. Given a WTS $\mathcal{M} = (S, \to, \ell)$, an equivalence relation $\mathcal{R} \subseteq S \times S$ on S is called a *generalised weighted bisimulation relation* if and only if for all $s, t \in S$, $s\mathcal{R}t$ implies

- (Atomic harmony) $\ell(s) = \ell(t)$,
- (Lower bound) $\theta^-(s)(T) = \theta^-(t)(T)$, and
- (Upper bound) $\theta^+(s)(T) = \theta^+(t)(T)$

for any $\mathcal{R}$-equivalence class $T \subseteq S$. ▲

Given $s, t \in S$ we say that s and t are generalized weighted bisimilar, written $s \sim t$, if and only if there exists a generalised weighted bisimulation relation $\mathcal{R}$ such that $s\mathcal{R}t$. We let $\sim$ denote generalised weighted bisimilarity which is defined as

$$\sim = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a generalised weighted bisimulation relation}\}.$$

We will now show that generalised weighted bisimilarity, $\sim$, is the largest generalised weighted bisimulation relation. To this end, we first need to show that $\sim$ is an equivalence relation.

Lemma A.2.8. *Generalised weighted bisimilarity, $\sim$, is an equivalence relation.*

Proof. In order to prove that generalised weighted bisimilarity is an equivalence relation, we have to show that it is reflexive, symmetric and transitive.

Reflexivity Consider the identity relation

$$\mathcal{I} = \{(s, s) \mid s \in S \text{ for some WTS } \mathcal{M} = (S, \to, \ell)\}.$$

It is trivial to verify that $\mathcal{I}$ is a generalised weighted bisimulation relation, and therefore $\mathcal{I} \subseteq \sim$.

Symmetry Let $\mathcal{M} = (S, \to, \ell)$ be a WTS and $s, t \in S$ states such that $s \sim t$. Because $s \sim t$ there must exist a generalised weighted bisimulation relation $\mathcal{R}$ such that $s\mathcal{R}t$. Let $\mathcal{R}' = \{(t, s) \mid (s, t) \in \mathcal{R}\}$. $\mathcal{R}'$ is clearly also a generalised weighted bisimulation relation implying $\mathcal{R}' \subseteq \sim$ and therefore $t \sim s$.

Transitivity Let $\mathcal{M} = (S, \to, \ell)$ be a WTS and $s, t, u \in S$ states such that $s \sim t$ and $t \sim u$. There must exist generalised weighted bisimulation relations $\mathcal{R}$ and $\mathcal{R}'$ such that $s\mathcal{R}t$ and $t\mathcal{R}'u$. Let $\mathcal{R}'' = (\mathcal{R} \cup \mathcal{R}')^+$ be the transitive closure of the union of $\mathcal{R}$ and $\mathcal{R}'$. Since $\mathcal{R}$ and $\mathcal{R}'$ are both equivalence relations, $\mathcal{R} \cup \mathcal{R}'$ is reflexive and symmetric, and since the transitive closure of a symmetric and reflexive relation is symmetric and reflexive, we get that $\mathcal{R}''$ is an equivalence relation. We need to show that $\mathcal{R}''$ is a generalised weighted bisimulation relation. Atomic harmony is trivially satisfied.

Suppose that $\theta(u)(T'') \neq \emptyset$ for some $T'' \in S/\mathcal{R}''$ implying the existence of a state $u' \in T''$ such that $\theta(u)(\{u'\}) \neq \emptyset$, further implying the existence of an equivalence class $T' \in S/\mathcal{R}'$ such that $u' \in T'$ and thus $\theta(u)(T') \neq \emptyset$. $t\mathcal{R}'u$ implies $\theta(t)(T') \neq \emptyset$ which further implies the existence of a state $t' \in T'$ such that $\theta(t)(\{t'\}) \neq \emptyset$. There must exist an equivalence class $T \in S/\mathcal{R}$ such that $t' \in T$ implying $\theta(t)(T) \neq \emptyset$. Because $s\mathcal{R}t$ we must have $\theta(s)(T) \neq \emptyset$ implying the existence of a state $s' \in T$ such that $\theta(s)(\{s'\}) \neq \emptyset$. $s', t' \in T$ implies $s'\mathcal{R}t'$, $t', u' \in T'$

implies $t'\mathcal{R}'u'$, and therefore $s'\mathcal{R}''u'$ implying $s' \in T''$ which further implies $\theta(s)(T'') \neq \emptyset$. Therefore $\theta(u)(T'') \neq \emptyset$ implies $\theta(s)(T'') \neq \emptyset$ for all $T'' \in S/\mathcal{R}''$. Symmetric arguments show that $\theta(s)(T'') \neq \emptyset$ implies $\theta(u)(T'') \neq \emptyset$ for all $T'' \in S/\mathcal{R}''$, and therefore $\theta(s)(T'') = \emptyset$ if and only if $\theta(u)(T'') = \emptyset$ for all $T'' \in S/\mathcal{R}''$.

Suppose towards a contradiction that $\theta^-(s)(T'') \neq \theta^-(u)(T'')$ for some $T'' \in S/\mathcal{R}''$. We have two cases to consider, namely

$$\theta^-(s)(T'') < \theta^-(u)(T'') \quad \text{and} \quad \theta^-(s)(T'') > \theta^-(u)(T'').$$

If $\theta^-(s)(T'') < \theta^-(u)(T'')$ there must exist a rational number $q \in \mathbb{Q}$ such that $\theta^-(s)(T'') < q < \theta^-(u)(T'')$, implying the existence of a state $s' \in T''$ such that $\theta^-(s)(T'') \leq \theta^-(s)(\{s'\}) < q$. There must exist $T \in S/\mathcal{R}$ such that $s' \in T$ implying $\theta^-(s)(T) < q$. Because $s\mathcal{R}t$ we must have $\theta^-(s)(T) = \theta^-(t)(T)$ implying the existence of a state $t' \in T$ such that $\theta^-(t)(\{t'\}) < q$. There must exist $T' \in S/\mathcal{R}'$ such that $t' \in T'$ implying $\theta^-(t)(T') < q$. Because $t\mathcal{R}'u$ we must have $\theta^-(t)(T') = \theta^-(u)(T')$ implying the existence of a state $u' \in T'$ such that $\theta^-(u)(\{u'\}) < q$. $s', t' \in T$ implies $s'\mathcal{R}t'$, $t', u' \in T'$ implies $t'\mathcal{R}u'$, and therefore $s'\mathcal{R}''u'$, implying $u' \in T''$ and therefore $\theta^-(u)(T'') < q$, leading to a contradiction. Symmetric arguments show that also $\theta^-(s)(T'') > \theta^-(u)(T'')$ leads to a contradiction and therefore $\theta^-(s)(T) = \theta^-(u)(T)$ for any $T \in S/\mathcal{R}''$.

Similar arguments show that $\theta^+(s)(T) = \theta^+(u)(T)$ for any $T \in S/\mathcal{R}''$ thus showing that $\mathcal{R}''$ is a generalised weighted bisimulation relation implying $\mathcal{R}'' \subseteq \sim$ and therefore $s \sim t$ and $t \sim u$ implies $s \sim u$. ■

Having established that $\sim$ is an equivalence relation, we will now show that it is indeed the largest generalised weighted bisimulation relation.

Theorem A.2.9. *Generalised weighted bisimilarity, $\sim$, is the largest generalised weighted bisimulation relation.*

Proof. We first show that $\sim$ is a generalised weighted bisimulation relation. By Lemma A.2.8 we know that $\sim$ is an equivalence relation. Let $\mathcal{M} = (S, \to, \ell)$ be a WTS and $s, t \in S$ states such that $s \sim t$. There must exist a generalised weighted bisimulation relation $\mathcal{R}$ such that $s\mathcal{R}t$, which trivially verifies atomic harmony.

Suppose that $\theta(t)(T) \neq \emptyset$ for some $T \in S/\sim$, implying the existence of a state $t' \in T$ such that $\theta(t)(\{t'\}) \neq \emptyset$. There must exist an equivalence class $T' \in S/\mathcal{R}$ such that $t' \in T'$, which implies that $\theta(t)(T') \neq \emptyset$. Because $s\mathcal{R}t$ we must have $\theta(s)(T') \neq \emptyset$, implying the existence of a state $s' \in T'$ such that $\theta(s)(\{s'\}) \neq \emptyset$. Because $s', t' \in T'$ we must have $s'\mathcal{R}t'$ and hence $s' \sim t'$, so $s' \in T$ and thus $\theta(s)(T) \neq \emptyset$. Symmetric arguments show that $\theta(s)(T) \neq \emptyset$

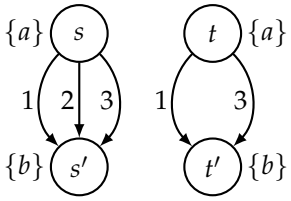


Figure A.2.2: $s \sim t$ but $s \not\sim_W t$.

implies $\theta(t)(T) \neq \emptyset$ and therefore $\theta(s)(T) = \emptyset$ if and only if $\theta(t)(T) = \emptyset$ for all $T \in S/\sim$.

Suppose $\theta^-(s)(T) \neq \theta^-(t)(T)$ for some $T \in S/\sim$. We have two cases to consider, namely $\theta^-(s)(T) < \theta^-(t)(T)$ and $\theta^-(s)(T) > \theta^-(t)(T)$. If $\theta^-(s)(T) < \theta^-(t)(T)$ there must exist a rational number $q \in \mathbb{Q}$ such that $\theta^-(s)(T) < q < \theta^-(t)(T)$, implying the existence of a state $s' \in T$ such that $\theta^-(s)(T) \leq \theta^-(s)(\{s'\}) < q$. There must exist $T' \in S/\mathcal{R}$ such that $s' \in T'$ and hence $\theta^-(s)(T') < q$. Because $s\mathcal{R}t$ we have $\theta^-(s)(T') = \theta^-(t)(T')$, which means that there exists a state $t' \in T'$ such that $\theta^-(t)(\{t'\}) < q$. $s', t' \in T'$ implies $s'\mathcal{R}t'$ which further implies $s' \sim t'$ and therefore $\theta^-(t)(T) < q$, leading to a contradiction. Symmetric arguments show that also $\theta^-(s)(T) > \theta^-(t)(T)$ leads to a contradiction, and therefore $\theta^-(s)(T) = \theta^-(t)(T)$ for all $T \in S/\sim$.

Similar arguments show that $\theta^+(s)(T) = \theta^+(t)(T)$ for any $T \in S/\sim$, thus showing that $\sim$ is a generalised weighted bisimulation relation.

$\sim$ was defined as the union of all generalised weighted bisimulation relations, so for any generalised weighted bisimulation relation $\mathcal{R}$ we must have $\mathcal{R} \subseteq \sim$, and hence we conclude that $\sim$ is the largest generalised weighted bisimulation relation. ■

In what follows, we will use bisimulation to mean generalised weighted bisimulation and bisimilarity to mean generalised weighted bisimilarity.

Example A.2.10. Consider the WTS depicted in Figure A.2.2. It is easy to see that $\{s', t'\}$ is a $\sim$-equivalence class, and in fact it is the only $\sim$-equivalence class with in-going transitions. Since $\theta^-(s)(\{s', t'\}) = \theta^-(t)(\{s', t'\}) = 1$ and $\theta^+(s)(\{s', t'\}) = \theta^+(t)(\{s', t'\}) = 3$ we must have $s \sim t$, but because $s \xrightarrow{2} s'$ and $t \not\xrightarrow{2}$ it cannot be the case that $s \sim_W t$. ♦

The following lemma shows that if two states are weighted bisimilar, then their image sets match exactly for any weighted bisimulation class.

Lemma A.2.11. *Let $\mathcal{M} = (S, \to, \ell)$ be a WTS and let $s, t \in S$. $s \sim_W t$ implies that $\theta(s)(T) = \theta(t)(T)$ for any $\sim_W$-equivalence class $T \subseteq S$.*

Proof. Assume $s \sim_W t$ and let $T \subseteq S$ be a $\sim_W$-equivalence class. If $r \in \theta(s)(T)$, then there exists some $s' \in T$ such that $s \xrightarrow{r} s'$. Because $s \sim_W t$, there must exist some $t' \in T$ such that $t \xrightarrow{r} t'$ and $s' \sim_W t'$. Since T is a $\sim_W$-equivalence class, this means that $r \in \theta(t)(T)$. A similar argument shows that if $r \in \theta(t)(T)$, then $r \in \theta(s)(T)$. ■

We can now show the following relationship between $\sim$ and $\sim_W$.

Theorem A.2.12. *Generalised weighted bisimilarity is coarser than weighted bisimilarity, i.e.*

$$\sim_W \subseteq \sim \quad \textit{and} \quad \sim_W \neq \sim .$$

Proof. Assume that $s \sim_W t$. We have that $\ell(s) = \ell(t)$, and by Lemma A.2.11, we have that $\theta(s)(T) = \theta(t)(T)$ for any $\sim_W$-equivalence class $T \subseteq S$. This implies in particular that $\theta^-(s)(T) = \theta^-(t)(T)$ and $\theta^+(s)(T) = \theta^+(t)(T)$. Hence $\sim_W$ is a bisimulation relation.

By Example A.2.10, the inclusion is strict. ■

This result is not surprising, as our bisimulation relation only looks at the extremes of the transition weights, whereas weighted bisimulation looks at all of the transition weights.

A.3 Logic

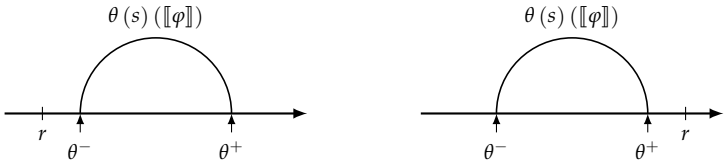
In this section we introduce a modal logic which is inspired by Markovian logic [30]. Our aim is that our logic should be able to capture the notion of bisimilar states as presented in the previous section, and as such it must be able to reason about the lower and upper bounds on transition weights.

Definition A.3.1. The formulas of the logic $\mathcal{L}$ are induced by the abstract syntax

$$\mathcal{L}: \quad \varphi, \psi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid L_r\varphi \mid M_r\varphi$$

where $r \in \mathbb{Q}_{\geq 0}$ is a non-negative rational number and $p \in \mathcal{AP}$ is an atomic proposition. ▲

L_r and M_r are modal operators. An illustration of how L_r and M_r are interpreted can be seen in Figure A.3.1. Intuitively, $L_r\varphi$ means that the cost of transitions to where φ holds is *at least* r (see Figure A.3.1a), and $M_r\varphi$ means that the cost of transitions to where φ holds is *at most* r (see Figure A.3.1b). We now give the precise semantics interpreted over WTSs.



(a) $\mathcal{M}, s \models L_r\varphi$ **(b)** $\mathcal{M}, s \models M_r\varphi$

Figure A.3.1: The semantics of L_r and M_r. If $\mathcal{M}, s \models L_r\varphi$, then r is to the left of $\theta^- (s) ([\![\varphi]\!])$, and if $\mathcal{M}, s \models M_r\varphi$, then r is to the right of $\theta^+ (s) ([\![\varphi]\!])$.

Definition A.3.2. Given a WTS $\mathcal{M} = (S, \to, \ell)$, a state $s \in S$ and a formula $\varphi \in \mathcal{L}$, the satisfiability relation $\models$ is defined inductively as

$$\begin{array}{lll} \mathcal{M}, s \models p & \text{iff} & p \in \ell(s), \\ \mathcal{M}, s \models \neg\varphi & \text{iff} & \mathcal{M}, s \not\models \varphi, \\ \mathcal{M}, s \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, s \models \varphi \text{ and } \mathcal{M}, s \models \psi, \\ \mathcal{M}, s \models L_r\varphi & \text{iff} & \theta^- (s) ([\![\varphi]\!]_{\mathcal{M}}) \geq r, \\ \mathcal{M}, s \models M_r\varphi & \text{iff} & \theta^+ (s) ([\![\varphi]\!]_{\mathcal{M}}) \leq r, \end{array}$$

where $[\![\varphi]\!]_{\mathcal{M}} = \{s \in S \mid \mathcal{M}, s \models \varphi\}$ is the set of all states of $\mathcal{M}$ having the property φ. ▲

We will omit the subscript $\mathcal{M}$ from $[\![\varphi]\!]_{\mathcal{M}}$ whenever the model is clear from the context. If $\mathcal{M}, s \models \varphi$ we say that $\mathcal{M}$ is a model of φ. A formula is said to be *satisfiable* if it has at least one model. We say that φ is a *validity* and write $\models \varphi$ if $\neg\varphi$ is not satisfiable. In addition to the operators defined by the syntax of $\mathcal{L}$, we also have the derived operators such as $\bot$, $\to$, etc. defined in the usual way. A *literal* is a formula that is of the form p or $\neg p$ where $p \in \mathcal{AP}$.

The formula $L_0\varphi$ has special significance in our logic, as this formula means that there exists some transition to where φ holds. In fact, it follows in a straightforward manner from the semantics that $\mathcal{M}, s \models L_0\varphi$ if and only if $\theta (s) ([\![\varphi]\!]) \neq \emptyset$. We can therefore encode the usual box and diamond modalities in our logic in the following way.

$$\Diamond\varphi = L_0\varphi \quad \Box\varphi = \neg\Diamond\neg\varphi.$$

Notice also that in general, the following schemes *do not hold*.

$$L_r\varphi \wedge L_r\psi \to L_r(\varphi \wedge \psi)$$
$$M_r\varphi \wedge M_r\psi \to M_r(\varphi \wedge \psi)$$

The reason that they do not hold in general is that there may be no transition to where $\varphi \wedge \psi$ holds, i.e. $\neg L_0(\varphi \wedge \psi)$. If we assume $L_0(\varphi \wedge \psi)$, then both

schemes hold, as we show in Lemma A.4.1. Another thing to note about the logic is that the formulas $L_r\varphi$ and $L_r\neg\varphi$ can both hold in the same model. To see this, simply construct a state that has two transitions with weight $x \geq r$ to two different states, one where φ holds and one where φ does not hold.

Example A.3.3. Consider again our model of a robot vacuum cleaner depicted in Figure A.2.1. Perhaps we want a guarantee that it takes no more than one time unit to go from a waiting state to a charging state. This can be expressed by the formula $\texttt{waiting} \rightarrow M_1\texttt{charging}$, but since we know the only waiting state in our model is s_1 this can be simplified to simply checking whether $\mathcal{M}, s_1 \models M_1\texttt{charging}$. We thus have to check that $\theta^+(s_1)([\![\texttt{charging}]\!]) \leq 1$. We do this by constructing the image set $\theta(s_1)([\![\texttt{charging}]\!])$. Since we have $[\![\texttt{charging}]\!] = \{s_3\}$, it follows that

$$\theta(s_1)([\![\texttt{charging}]\!]) = \{1, 2\}.$$

Hence

$$\theta^+(s_1)([\![\texttt{charging}]\!]) = 2 \not\leq 1,$$

so $\mathcal{M}, s_1 \not\models M_1\texttt{charging}$. ♦

Lemma A.3.4. *Let $\mathcal{M} = (S, \rightarrow, \ell)$ be an image-finite WTS and $s \in S$. Let $T \subseteq S$ be a set such that all elements of T satisfy exactly the same formulas, and furthermore for any $t \in T$ and $t' \notin T$, there exists a formula φ such that $t \models \varphi$ and $t' \not\models \varphi$. Then there exists a formula $\varphi \in \mathcal{L}$ such that $\theta(s)(T) = \theta(s)([\![\varphi]\!])$.*

Proof. The idea of the proof is to repeatedly use the observation that if $t' \notin T$, then there exists a formula φ such that $t' \not\models \varphi$ and $t \models \varphi$ for all $t \in T$. First pick some formula φ_1 such that $t \models \varphi_1$ for all $t \in T$. Then $T \subseteq [\![\varphi_1]\!]$, so $\theta(s)(T) \subseteq \theta(s)([\![\varphi_1]\!])$. If $\theta(s)(T) \subsetneq \theta(s)([\![\varphi_1]\!])$, then there must exist some $t_1 \notin T$ such that $s \xrightarrow{r} t_1$ and $t_1 \models \varphi_1$. Since $t_1 \notin T$, there must exist some formula φ_2 such that $t_1 \not\models \varphi_2$ and $t \models \varphi_2$ for all $t \in T$. We then get $\theta(s)(T) \subseteq \theta(s)([\![\varphi_1 \wedge \varphi_2]\!])$. Again, if $\theta(s)(T) \subsetneq \theta(s)([\![\varphi_1 \wedge \varphi_2]\!])$, then there must exist some $t_2 \notin T$ such that $s \xrightarrow{r} t_2$ and $t_2 \models \varphi_2$. Since $t_2 \notin T$, there must exist some formula φ_3 such that $t_1 \not\models \varphi_3$ and $t \models \varphi_3$ for all $t \in T$. Since $\mathcal{M}$ is image-finite, there can only be finitely many states $t_i \notin T$ with $s \xrightarrow{r} t_i$, so continuing in the same way, we will eventually get a formula $\varphi_1 \wedge \cdots \wedge \varphi_n$ such that $\theta(s)(T) = \theta(s)([\![\varphi_1 \wedge \cdots \wedge \varphi_n]\!])$. ■

Next we show that our logic $\mathcal{L}$ is invariant under bisimulation, which is also known as the Hennessy-Milner property. In order to prove this result, we have to restrict our models to only those that are image-finite, as shown by the following example.

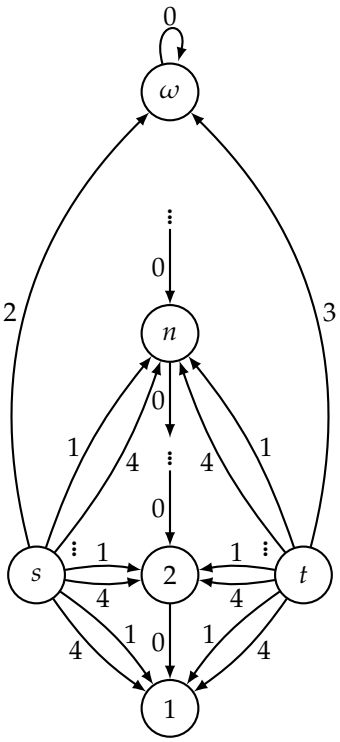


Figure A.3.2: s and t satisfy the same logical formulas, but $s \not\sim t$.

Example A.3.5. Consider the WTS depicted in Figure A.3.2 with state space $S = \mathbb{N} \cup \{\omega, s, t\}$ and $\ell(s') = \emptyset$ for all $s' \in S$. The transition relation is given by $\omega \xrightarrow{0} \omega$, $s \xrightarrow{2} \omega$, $t \xrightarrow{3} \omega$, and $n+1 \xrightarrow{0} n$, $s \xrightarrow{1} n$, and $t \xrightarrow{1} n$ for all $n \in \mathbb{N}$.

Then we have that $s_1 \sim s_2$ if and only if $s_1 = s_2$, since any states in $\mathbb{N} \cup \{\omega\}$ can be distinguished by the number of steps they can take, and s and t can be distinguished by the fact that $\theta^-(s)(\{\omega\}) = 2 \neq 3 = \theta^-(t)(\{\omega\})$. However, s and t satisfy all the same formulas, since any formula that holds in ω will also hold in n for some $n \in \mathbb{N}$, and the weights on the transitions to ω will therefore be masked by the bounds 1 and 4, and hence any formula can not distinguish between s and t. ⧫

The proof strategy follows a classical pattern: The left to right direction is shown by induction on φ for $\varphi \in \mathcal{L}$. The right to left direction is shown by constructing a relation $\mathcal{R}$ relating those states that satisfy the same formulas and showing that this relation is a bisimulation relation.

Theorem A.3.6 (Bisimulation invariance)**.** *For any WTS $\mathcal{M} = (S, \to, \ell)$ and states $s, t \in S$ it holds that*

$$s \sim t \quad \textit{implies} \quad [\forall \varphi \in \mathcal{L}.\ \mathcal{M}, s \models \varphi \ \textit{ iff } \ \mathcal{M}, t \models \varphi]\,.$$

Furthermore, if $\mathcal{M}$ is image-finite, then it also holds that

$$[\forall \varphi \in \mathcal{L}.\ \mathcal{M}, s \models \varphi \ \textit{ iff } \ \mathcal{M}, t \models \varphi] \quad \textit{implies} \quad s \sim t.$$

Proof. We first show that $s \sim t$ implies $\mathcal{M}, s \models \varphi$ if and only if $\mathcal{M}, t \models \varphi$ for all $\varphi \in \mathcal{L}$ by induction on φ. The Boolean cases are trivial. If $\varphi = L_r\psi$, then we have $\theta^-(s)([\![\psi]\!]) \geq r$, which implies that $\theta^-(s)([\![\psi]\!]) \neq -\infty$. Assume towards a contradiction that $\theta^-(t)([\![\psi]\!]) < r$. It can not be the case that $\theta^-(t)([\![\psi]\!]) = -\infty$, hence it follows that $[\![\psi]\!]$ and $\theta(t)([\![\psi]\!])$ are non-empty, so there must exist some element $t' \in [\![\psi]\!]$ such that $\theta^-(t)([\![\psi]\!]) \leq \theta^-(t)(\{t'\}) < r$. Since $\sim$ is an equivalence relation, there must exists some $\sim$-equivalence class T such that $t' \in T$. This means that $\{t'\} \subseteq T$, so that also $\theta^-(t)(T) \leq \theta^-(t)(\{t'\}) < r$. By the induction hypothesis we have that $T \subseteq [\![\psi]\!]$. Because $s \sim t$, we have that $\theta^-(s)(T) = \theta^-(t)(T) < r$, so by monotonicity we get $\theta^-(s)([\![\psi]\!]) \leq \theta^-(s)(T) < r$, which is a contradiction. The M_r case is handled similarly.

For the reverse direction, assume that $\mathcal{M}$ is image-finite. We have to show that if for all $\varphi \in \mathcal{L}$, $\mathcal{M}, s \models \varphi$ if and only if $\mathcal{M}, t \models \varphi$ then $s \sim t$. To this end, we define a relation $\mathcal{R}$ on S as

$$\mathcal{R} = \{(s,t) \in S \times S \mid \forall \varphi \in \mathcal{L}.\ \mathcal{M}, s \models \varphi \text{ iff } \mathcal{M}, t \models \varphi\}\,.$$

$\mathcal{R}$ is clearly an equivalence relation and $s\mathcal{R}t$.

It is clear that $\ell(s) = \ell(t)$. Next we show that $\theta^-(s)(T) = \theta^-(t)(T)$ and $\theta^+(s)(T) = \theta^+(t)(T)$ for any $\mathcal{R}$-equivalence class T. Let $T \subseteq S$ be an $\mathcal{R}$-equivalence class. We first show that $\theta(s)(T) = \emptyset$ if and only if $\theta(t)(T) = \emptyset$. Assume that $\theta(s)(T) = \emptyset$. By Lemma A.3.4 there exists a formula φ such that $\theta(s)(T) = \theta(s)([\![\varphi]\!]) = \emptyset$, and therefore $s \not\models L_0\varphi$. Now assume towards a contradiction that $\theta(t)(T) \neq \emptyset$. Since $\mathcal{M}$ is image-finite, there must be a finite subset $T' \subseteq T$ such that $\theta(t)(T) = \theta(t)(T')$. By Lemma A.2.5, we then get $\theta(t)(T) = \bigcup_{t' \in T'} \theta(t)(\{t'\}) \neq \emptyset$, from which it follows that there must be some $t' \in T'$ such that $\theta(t)(\{t'\}) \neq \emptyset$. Since $t' \in T$, we must have $t' \models \varphi$, and therefore $t \models L_0\varphi$, which contradicts the fact that $s\mathcal{R}t$ and $s \not\models L_0\varphi$.

Now assume that $\theta(s)(T) \neq \emptyset$ and $\theta(t)(T) \neq \emptyset$. We need to show that $\theta^-(s)(T) = \theta^-(t)(T)$ and $\theta^+(s)(T) = \theta^+(t)(T)$. We do this by contradiction, which gives us four cases to consider: $\theta^-(s)(T) < \theta^-(t)(T)$, $\theta^-(s)(T) > \theta^-(t)(T)$, $\theta^+(s)(T) < \theta^+(t)(T)$, and $\theta^+(s)(T) > \theta^+(t)(T)$.

For the case of $\theta^-(s)(T) < \theta^-(t)(T)$, there exists $q \in \mathbb{Q}_{\geq 0}$ such that

$$\theta^-(s)(T) < q < \theta^-(t)(T).$$

By Lemma A.3.4, there exists a formula φ such that $\theta^-(t)(T) = \theta^-(t)([\![\varphi]\!])$. Since $T \subseteq [\![\varphi]\!]$, we then obtain

$$\theta^-(s)([\![\varphi]\!]) \leq \theta^-(s)(T) < q < \theta^-(t)(T) = \theta^-(t)([\![\varphi]\!]),$$

which implies that $s \not\models L_q\varphi$ but $t \models L_q\varphi$, and thus we get a contradiction. The other cases are handled similarly. ■

A.4 Metatheory

In this section we propose an axiomatisation for our logic that we prove not only sound, but also complete with respect to the proposed semantics.

A.4.1 Axiomatic System

Let $r, s \in \mathbb{Q}_{\geq 0}$. Then the deducibility relation $\vdash \subseteq 2^{\mathcal{L}} \times \mathcal{L}$ is a classical conjunctive deducibility relation, and is defined as the smallest relation which satisfies the axioms of propositional logic in addition to the axioms given in Table A.4.1. We will write $\vdash \varphi$ to mean $\emptyset \vdash \varphi$, and we say that a formula or a set of formulas is *consistent* if it can not derive $\bot$.

The axioms presented in Table A.4.1 bear some resemblance to the axiomatic systems of [30] and [5]. Notably, our axiom A2 is almost identical to A2 of these works and capture similar properties about the systems being studied, with the major difference being that we reason about transition weights whereas the aforementioned works reason about rates or probabilities of transitions. Also worth noting here is the similarity between the rule

(A1):	$\vdash \neg L_0 \bot$	
(A2):	$\vdash L_{r+q}\varphi \to L_r\varphi$	if $q > 0$
(A2′):	$\vdash M_r\varphi \to M_{r+q}\varphi$	if $q > 0$
(A3):	$\vdash L_r\varphi \wedge L_q\psi \to L_{\min\{r,q\}}(\varphi \vee \psi)$	
(A3′):	$\vdash M_r\varphi \wedge M_q\psi \to M_{\max\{r,q\}}(\varphi \vee \psi)$	
(A4):	$\vdash L_r(\varphi \vee \psi) \to L_r\varphi \vee L_r\psi$	
(A5):	$\vdash \neg L_0\psi \to (L_r\varphi \to L_r(\varphi \vee \psi))$	
(A5′):	$\vdash \neg L_0\psi \to (M_r\varphi \to M_r(\varphi \vee \psi))$	
(A6):	$\vdash L_{r+q}\varphi \to \neg M_r\varphi$	if $q > 0$
(A7):	$\vdash M_r\varphi \to L_0\varphi$	
(R1):	$\vdash \varphi \to \psi \implies \vdash (L_r\psi \wedge L_0\varphi) \to L_r\varphi$	
(R1′):	$\vdash \varphi \to \psi \implies \vdash (M_r\psi \wedge L_0\varphi) \to M_r\varphi$	
(R2):	$\vdash \varphi \to \psi \implies \vdash L_0\varphi \to L_0\psi$	

Table A.4.1: The axioms for our axiomatic system, where $\varphi, \psi \in \mathcal{L}$ and $q, r \in \mathbb{Q}$.

R1 of these works and R1 of our axiomatic system. A notable difference is that we do not have the additive properties of measures for disjoint sets (since we are not working with probability measures), as is captured by the axioms A3 and A4 of these works. Also, in one of the axiomatisations of [30], the axioms A2 and A2′ are not axioms, but can be derived from the axioms.

Rules R2 and R3 of [30] and [5] reflect the Archimedean property of rationals, and while similar axioms can be proven sound in our setting, these were not needed to show our completeness result. We suspect, however, that if we were to pursue strong completeness, infinitary axioms similar to these would be needed.

Axiom A1 captures the notion that since $\bot$ is never satisfied, we can never take a transition to where $\bot$ holds. Axiom A2 says that if we know some value is the lower bound for going to where φ holds, then any lower value is also a lower bound for going to where φ holds. Axiom A2′ is the analogue for upper bounds. Axioms A3-A4 show how L_r and M_r distribute over conjunction and disjunction. The version of axiom A4 where L_r is replaced with M_r is also sound, but as we show in Lemma A.4.1, it can be proven from the other axioms. Axioms A5 and A5′ say that if it is not possible to take a transition to where ψ holds, then including the states where ψ holds does not change the bounds. Axioms A6 and A7 show the relationship between L_r and M_r. In particular, A6 ensures that all bounds are well-formed. Notice also that the contrapositive of axiom A2 and A7 together gives us that $\neg L_0\varphi$ implies $\neg L_r\varphi$ and $\neg M_r\varphi$ for any $r \in \mathbb{Q}_{\geq 0}$. The rules R1 and R1′ give a sort of monotonicity for L_r and M_r, and rule R2 says that if ψ follows from φ, then if it is possible to take a transition to where φ holds, it is also possible to take a transition to where ψ holds.

We now show some of the theorems which can be deduced from the axioms. T1, T1′, and T5 together complete the distributivity properties for conjunction and disjunction. T2 and T2′ make precise the intuitively clear idea that if two formulas are equivalent, then their upper and lower bounds should also be the same. T3 extends axiom A1 to hold for any $r \geq 0$, and T4 then extends this to any φ which implies $\bot$.

Lemma A.4.1. *From the axioms listed in Table A.4.1 we can derive the following theorems:*

(T1): $\vdash (L_r\varphi \wedge L_q\psi \wedge L_0(\varphi \wedge \psi)) \to L_{\max\{r,q\}}(\varphi \wedge \psi)$
(T1′): $\vdash (M_r\varphi \wedge M_q\psi \wedge L_0(\varphi \wedge \psi)) \to M_{\min\{r,q\}}(\varphi \wedge \psi)$
(T2): $\vdash \varphi \leftrightarrow \psi \implies \vdash L_r\varphi \leftrightarrow L_r\psi$
(T2′): $\vdash \varphi \leftrightarrow \psi \implies \vdash M_r\varphi \leftrightarrow M_r\psi$
(T3): $\vdash \neg L_r\bot, \quad r \geq 0$
(T4): $\vdash \varphi \to \bot \implies \vdash \neg L_r\varphi, \quad r \geq 0$
(T5): $\vdash M_r(\varphi \vee \psi) \to M_r\varphi \vee M_r\psi$

Proof.

T1 Rule R1 implies

$$\vdash \neg L_q(\varphi \wedge \psi) \to (\neg L_q\varphi \vee \neg L_0(\varphi \wedge \psi)),$$

so also

$$\vdash \neg L_q(\varphi \wedge \psi) \to (\neg L_q\varphi \vee \neg L_0(\varphi \wedge \psi) \vee \neg L_r\psi).$$

This is equivalent to

$$\vdash (L_r\varphi \wedge L_q\psi \wedge L_0(\varphi \wedge \psi)) \to L_q(\varphi \wedge \psi).$$

T1′ Similar to T1.

T2 Suppose $\vdash \varphi \leftrightarrow \psi$. We have that $\vdash L_r\varphi \to L_0\varphi$ by A2 and $\vdash L_0\varphi \to L_0\psi$ by R2. Hence $\vdash L_r\varphi \to (L_r\varphi \wedge L_0\psi)$, so $\vdash L_r\varphi \to L_r\psi$ by R1. A similar argument shows that $\vdash L_r\psi \to L_r\varphi$, so $\vdash L_r\varphi \leftrightarrow L_r\psi$.

T2′ Similar to T2.

T3 From axiom A1 we know that $\vdash \neg L_0\bot$ which, by the contrapositive of A2, implies $\vdash \neg L_r\bot$ for any $r > 0$.

T4 Suppose $\vdash \varphi \to \bot$. We know for any $\psi \in \mathcal{L}$ that $\vdash \bot \to \psi$ and therefore $\vdash \varphi \to \bot \implies \vdash \varphi \leftrightarrow \bot$. From A1 we know that $\vdash \neg L_0\bot$ and from T3 that $\vdash \neg L_r\bot$ for any $r > 0$ implying, by T2, that $\vdash \neg L_r\varphi$ for any $r \geq 0$.

T5 By axiom A7 we get $\vdash M_r(\varphi \vee \psi) \to L_0(\varphi \vee \psi)$ and A4 gives

$$\vdash L_0(\varphi \vee \psi) \to L_0\varphi \vee L_0\psi.$$

Hence we get $\vdash M_r(\varphi \vee \psi) \to (M_r(\varphi \vee \psi) \wedge L_0\varphi) \vee (M_r(\varphi \vee \psi) \wedge L_0\psi)$. Since $\vdash \varphi \to (\varphi \vee \psi)$ and $\vdash \psi \to (\varphi \vee \psi)$, rule R1′ then gives

$$\vdash M_r(\varphi \vee \psi) \to M_r\varphi \vee M_r\psi. \qquad \blacksquare$$

Next we prove that our axioms are indeed sound.

Theorem A.4.2 (Soundness)**.**

$$\vdash \varphi \quad \textit{implies} \quad \models \varphi.$$

Proof. The soundness of each axiom is easy to show, and many of them use the distributive property from Lemma A.2.5. Here we prove the soundness for a few of the more interesting axioms.

A3 Suppose $\mathcal{M}, s \models L_r\varphi \wedge L_q\psi$ implying that $\mathcal{M}, s \models L_r\varphi$ and $\mathcal{M}, s \models L_q\psi$, implying further that $\theta^-(s)([\![\varphi]\!]) \geq r$ and $\theta^-(s)([\![\psi]\!]) \geq q$.

By Lemma A.2.5 we must have that

$$\theta(s)([\![\varphi \vee \psi]\!]) = \theta(s)([\![\varphi]\!] \cup [\![\psi]\!]) = \theta(s)([\![\varphi]\!]) \cup \theta(s)([\![\psi]\!])$$

and because $\theta^-(s)([\![\varphi]\!]) \geq r$ and $\theta^-(s)([\![\psi]\!]) \geq q$ we must have

$$\theta^-(s)([\![\varphi \vee \psi]\!]) = \inf \theta(s)([\![\varphi]\!]) \cup \theta(s)([\![\psi]\!]) \geq \min\{r, q\}$$

implying $\mathcal{M}, s \models L_{\min\{r,q\}}(\varphi \vee \psi)$.

A4 Suppose $\mathcal{M}, s \models L_r(\varphi \vee \psi)$ implying that

$$\theta^-(s)([\![\varphi \vee \psi]\!]) = \inf \theta(s)([\![\varphi]\!]) \cup \theta(s)([\![\psi]\!]) \geq r.$$

This implies that at least one of $\theta(s)([\![\varphi]\!])$ and $\theta(s)([\![\psi]\!])$ is non-empty. If $\theta(s)([\![\varphi]\!]) \neq \emptyset$, then $\theta^-(s)([\![\varphi]\!]) \geq r$, and also if $\theta(s)([\![\psi]\!]) \neq \emptyset$, then $\theta^-(s)([\![\psi]\!]) \geq r$, so at least one of $\mathcal{M}, s \models L_r\varphi$ and $\mathcal{M}, s \models L_r\psi$ must hold. Hence $\mathcal{M}, s \models L_r\varphi \vee L_r\psi$.

A6 Suppose $\mathcal{M}, s \models L_{r+q}\varphi$ implying that

$$\theta^-(s)([\![\varphi]\!]) = \inf \theta(s)([\![\varphi]\!]) \geq r + q.$$

It is clear that $\inf \theta(s)([\![\varphi]\!]) \leq \sup \theta(s)([\![\varphi]\!])$, so

$$\theta^+(s)([\![\varphi]\!]) = \sup \theta(s)([\![\varphi]\!]) \geq \inf \theta(s)([\![\varphi]\!]) \geq r + q > r.$$

Therefore, it cannot be the case that $\mathcal{M}, s \models M_r\varphi$ and thus $\mathcal{M}, s \models \neg M_r\varphi$.

R1 Suppose $\models \varphi \to \psi$ implying that $[\![\varphi]\!] \subseteq [\![\psi]\!]$, implying further, by the monotonicity of θ, that $\theta(s)([\![\varphi]\!]) \subseteq \theta(s)([\![\psi]\!])$. Suppose further that $\mathcal{M}, s \models L_r\psi \wedge L_0\varphi$ implying $\mathcal{M}, s \models L_r\psi$ and $\mathcal{M}, s \models L_0\varphi$, implying further that

$$\theta^-(s)([\![\psi]\!]) = \inf \theta(s)([\![\psi]\!]) \geq r \quad \text{and} \quad \theta(s)([\![\varphi]\!]) \neq \emptyset.$$

Since $\theta(s)([\![\varphi]\!])$ is non-empty, we then get that

$$\inf \theta(s)([\![\varphi]\!]) \geq \inf \theta(s)([\![\psi]\!]) \geq r,$$

which means that $\mathcal{M}, s \models L_r\varphi$. ■

A.4.2 Finite Model Property and Completeness

With our axiomatisation proven sound we are now ready to present our main results, namely that our logic has the finite model property and that our axiomatisation is complete.

To show the finite model property we will adapt the classical filtration method to our setting. Starting from an arbitrary formula ρ, we define a finite fragment of our logic, $\mathcal{L}[\rho]$, which we then use to construct a finite model for ρ. The main difference from the classical filtration method is that we must find an upper and a lower bound for the transitions in the model. For an arbitrary formula $\rho \in \mathcal{L}$ we define the following based on ρ:

- Let $Q_\rho \subseteq \mathbb{Q}_{\geq 0}$ be the set of all rational numbers $r \in \mathbb{Q}_{\geq 0}$ such that L_r or M_r appears in the syntax of ρ.
- Let Σ_ρ be the set of all atomic propositions $p \in \mathcal{AP}$ such that p appears in the syntax of ρ.
- The *granularity* of ρ, denoted as $gr(\rho)$, is the least common denominator of all the elements in Q_ρ.
- The *range* of ρ, denoted as R_ρ, is defined as

$$R_\rho = \begin{cases} \emptyset & \text{if } Q_\rho = \emptyset \\ I_\rho \cup \{0\} & \text{otherwise,} \end{cases}$$

where $I_\rho = \left\{ q \in \mathbb{Q}_{\geq 0} \mid \exists j \in \mathbb{N}.\ q = \frac{j}{gr(\rho)} \text{ and } \min Q_\rho \leq q \leq \max Q_\rho \right\}$. Here the granularity is used to pick out finitely many numbers in the interval. Note that we need to add 0 to R_ρ whether or not ρ actually contains 0 in any of its modalities. This is because, as we have pointed out before, formulas involving L_0 have special significance in our logic.

- The *modal depth* of ρ, denoted as $md(\rho)$, is defined inductively as:

$$md(\rho) = \begin{cases} 0 & \text{if } \rho = p \in \mathcal{AP} \\ md(\varphi) & \text{if } \rho = \neg\varphi \\ \max\{md(\varphi_1), md(\varphi_2)\} & \text{if } \rho = \varphi_1 \wedge \varphi_2 \\ 1 + md(\varphi) & \text{if } \rho = L_r\varphi \text{ or } \rho = M_r\varphi. \end{cases}$$

Since all formulas are finite, the modal depth is always a non-negative integer. The *language* of ρ, denoted by $\mathcal{L}[\rho]$, is defined as

$$\mathcal{L}[\rho] = \{\varphi \in \mathcal{L} \mid R_\varphi \subseteq R_\rho, md(\varphi) \leq md(\rho) \text{ and } \Sigma_\varphi \subseteq \Sigma_\rho\},$$

and we take $\mathcal{L}_\leftrightarrow[\rho]$ to be the Lindenbaum algebra of $\mathcal{L}[\rho]$, i.e. the quotient with respect to logical equivalence. The Lindenbaum algebra is a Boolean algebra with equivalence classes as elements. Note that the quotient

$$h : \mathcal{L}[\rho] \to \mathcal{L}_\leftrightarrow[\rho]$$

is a homomorphism between Boolean algebras, and therefore preserves the structure of $\mathcal{L}[\rho]$. For each element $x \in \mathcal{L}_\leftrightarrow[\rho]$, we fix now a formula $\varphi \in x$ to be the representative of that equivalence class, and we write $\hat{\varphi}$ for x. The order $\leq$ in $\mathcal{L}_\leftrightarrow[\rho]$ is then given by $\hat{\varphi} \leq \hat{\psi}$ if and only if $\vdash \varphi \to \psi$. The join and meet in $\mathcal{L}_\leftrightarrow[\rho]$ are given by

$$\hat{\varphi} \vee \hat{\psi} = h(\varphi \vee \psi) \quad \hat{\varphi} \wedge \hat{\psi} = h(\varphi \wedge \psi),$$

and complement is given by

$$\neg\hat{\varphi} = h(\neg\varphi).$$

Note here the difference between $h(\varphi)$ and $\hat{\varphi}$. The quotient h sends φ to its equivalence class $x \in \mathcal{L}_\leftrightarrow[\rho]$. However, it may be the case that φ is not the representative for x, but some other formula ψ is. In that case we have $h(\varphi) = x = \hat{\psi}$. On the other hand, $\hat{\varphi}$ denotes both that $\varphi \in \hat{\varphi}$, and also that φ is the chosen representative of its equivalence class, which ensures that in this case we have $h(\varphi) = \hat{\varphi}$.

The idea is that Σ_ρ ensures that only finitely many atomic propositions are used, R_ρ ensures that only finitely many weights on the modalities are used, and $md(\rho)$ puts a bound on the modal depth of formulas. The language $\mathcal{L}[\rho]$ itself is not finite, but contains only finitely many logically non-equivalent formulas. Hence $\mathcal{L}_\leftrightarrow[\rho]$ must be finite, and as we shall see, it contains all the information necessary to construct a model for ρ.

Proposition A.4.3. *The language $\mathcal{L}_\leftrightarrow[\rho]$ is finite.*

Proof. Let $\mathcal{L}^n_{\leftrightarrow}[\rho]$ be the subset of $\mathcal{L}_{\leftrightarrow}[\rho]$ which only contains formulas of modal depth n. Then it is clear that

$$\mathcal{L}_{\leftrightarrow}[\rho] = \bigcup_{i=0}^{md(\rho)} \mathcal{L}^i_{\leftrightarrow}[\rho].$$

We will now prove by induction on the modal depth that for each i, $\mathcal{L}^i_{\leftrightarrow}[\rho]$ is finite.

$i = 0$: In this case, each element of $\mathcal{L}^0_{\leftrightarrow}[\rho]$ is a Boolean combination of atomic propositions in Σ_ρ. There are $2^{2^{|\Sigma_\rho|}}$ non-equivalent such formulas, so this set is finite.

$i > 0$: Each element of $\mathcal{L}^i_{\leftrightarrow}[\rho]$ is a Boolean combination of formulas of the form $L_r\varphi$ and $M_r\varphi$, where $\varphi \in \mathcal{L}^j_{\leftrightarrow}[\rho]$ for some $j < i$ and $r \in R_\rho$. By induction hypothesis, we know that there are only finitely many such φ. We know from Lemma A.4.1 that if φ and ψ are logically equivalent, then $L_r\varphi$ and $L_r\psi$ as well as $M_r\varphi$ and $M_r\psi$ are also logically equivalent. Since R_ρ is finite, we conclude that $\mathcal{L}^i_{\leftrightarrow}[\rho]$ is finite. ■

In order to define the model, we need the standard notions of filters and ultrafilters on Boolean algebras [15]. A non-empty subset of a Boolean algebra B is called a *filter* if it is upward-closed with respect to the order, and closed under finite meets. A filter F is *proper* if $F \neq B$. An *ultrafilter* is a proper filter which is maximal in the sense of set inclusion.

The following property of ultrafilters is often useful.

Lemma A.4.4. *For an ultrafilter F of $\mathcal{L}_{\leftrightarrow}[\rho]$ it holds that for any $\varphi \in \mathcal{L}[\rho]$, either $h(\varphi) \in F$ or $\neg h(\varphi) \in F$, but not both.*

We let $\mathcal{U}[\rho]$ denote the set of all ultrafilters on $\mathcal{L}_{\leftrightarrow}[\rho]$. Since $\mathcal{L}_{\leftrightarrow}[\rho]$ is finite, $\mathcal{U}[\rho]$ is also finite and consequently, any ultrafilter $u \in \mathcal{U}[\rho]$ must be a finite set. For any set $\Phi \subseteq \mathcal{L}_{\leftrightarrow}[\rho]$, the characteristic formula of Φ, denoted $(\!|\Phi|\!)$, is defined as

$$(\!|\Phi|\!) = \bigwedge_{\hat{\varphi} \in \Phi} \varphi.$$

Note that $(\!|\Phi|\!) \in \mathcal{L}[\rho]$ is a finite formula, and that if $u \in \mathcal{U}[\rho]$, then $h((\!|u|\!)) \in u$.

We will now construct a (finite) model, $\mathcal{M}_\rho$, for ρ with state space $\mathcal{U}[\rho]$. In order to define the transition relation $\rightarrow_\rho \subseteq \mathcal{U}[\rho] \times \mathbb{R}_{\geq 0} \times \mathcal{U}[\rho]$, we consider any two ultrafilters $u, v \in \mathcal{U}[\rho]$ and define two functions

$$L, M : \mathcal{U}[\rho] \times \mathcal{U}[\rho] \to 2^{R_\rho}$$

as

$$L(u,v) = \{r \mid h(L_r(\!|v|\!)) \in u\} \quad \text{and} \quad M(u,v) = \{s \mid h(M_s(\!|v|\!)) \in u\}.$$

The following lemma establishes a relationship between L and M, that we will need to define the transition relation. The lemma is a straightforward consequence of axiom $A7$.

Lemma A.4.5. *Given any ultrafilters $u, v \in \mathcal{U}[\rho]$, it can not be the case that $L(u,v) = \emptyset$ and $M(u,v) \neq \emptyset$.*

Proof. Assume towards a contradiction that $L(u,v) = \emptyset$ and $M(u,v) \neq \emptyset$. Then there exists some $r \in Q_\rho$ such that

$$h(\neg L_0(\!|v|\!)) \in u \quad \text{and} \quad h(M_r(\!|v|\!)) \in u.$$

However, by axiom A7, this implies that $h(L_0(\!|v|\!)) \in u$, which is a contradiction. ■

We can now define the transition relation in terms of $L(u,v)$ and $M(u,v)$. In Figure A.4.1, we have illustrated the different cases that we must consider. Here, the area between $\min Q_\rho$ and $\max Q_\rho$ is the only part that the restricted language $\mathcal{L}[\rho]$ can speak about. The arches represent the interval within which transitions with that weight are possible. For any of the arches in the figure, we have the following correspondence with L_r and M_r.

- If a number r on the real line is contained within the arch, then we have $h(\neg L_r(\!|v|\!)) \in u$ and $h(\neg M_r(\!|v|\!)) \in u$.
- If a number r on the real line is to the left of the arch, then we have $h(L_r(\!|v|\!)) \in u$ and $h(\neg M_r(\!|v|\!)) \in u$.
- If a number r on the real line is to the right of the arch, then we have $h(M_r(\!|v|\!)) \in u$ and $h(\neg L_r(\!|v|\!)) \in u$.

In case (a) in Figure A.4.1, we therefore have $L(u,v) \neq \emptyset$ and $M(u,v) \neq \emptyset$, so we have all the information we need to define the transition. In case (b) and (f), we have $L(u,v) \neq \emptyset$ and $M(u,v) = \emptyset$, since there exist numbers within the interval $[\min Q_\rho, \max Q_\rho]$ that are to the left of these arches, but none that are to the right. This means that we have enough information to define the minimum transition, but we do not know what the maximum transition is. Note that we can not simply say that the maximum transition is $\max Q_\rho$, because that would imply $h(M_{\max Q_\rho}(\!|v|\!)) \in u$, but we know that $M(u,v) = \emptyset$. Hence we need to pick a number that is to the right of $\max Q_\rho$ as the maximum. In case (d), we have both $L(u,v) = \emptyset$ and $M(u,v) = \emptyset$. This implies that $h(\neg L_0(\!|v|\!)) \in u$, which means that there should be no transition from u to v. In case (c) and (e), we have $L(u,v) = \emptyset$ and $M(u,v) \neq \emptyset$, but according to Lemma A.4.5 these cases can never occur.

We therefore distinguish the following three cases in order to define the transition relation:

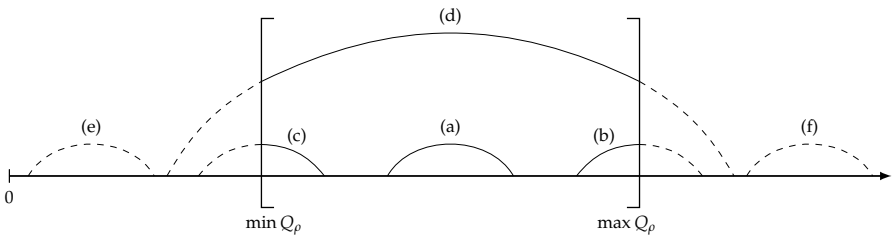


Figure A.4.1: When constructing a transition from u to v, we will only have information about what happens in the region Q_ρ and at 0. The line represents the non-negative real line and the arches represent the transitions that would be possible in a full model (i.e. one not restricted to $\mathcal{L}[\rho]$). The dashed part of the arches represent the part of the transition that we do not have information about.

1. If $L(u,v) \neq \emptyset$ and $M(u,v) \neq \emptyset$, then we add the two transitions $u \xrightarrow{r_1} v$ and $u \xrightarrow{r_2} v$ where $r_1 = \max L(u,v)$ and $r_2 = \min M(u,v)$.
2. If $L(u,v) \neq \emptyset$ and $M(u,v) = \emptyset$, then we add the two transitions $u \xrightarrow{r_1} v$ and $u \xrightarrow{r_2} v$ where $r_1 = \max L(u,v)$ and $r_2 = \max Q_\rho + \frac{1}{gr(\rho)}$.
3. If $L(u,v) = \emptyset$ and $M(u,v) = \emptyset$, then there is no transition from u to v.

The following lemma tells us that these transitions are well-formed, i.e. that the lower bound on transitions is less than or equal to the upper bound.

Lemma A.4.6. *For any ultrafilters $u, v \in \mathcal{U}[\rho]$, if $L(u,v) \neq \emptyset$ and $M(u,v) \neq \emptyset$, then* $\max L(u,v) \leq \min M(u,v)$.

Proof. Assume towards a contradiction that $\max L(u,v) > \min M(u,v)$. Then there exist $q, q' \in Q_\rho$ such that $q > q'$, $h(L_q(\!|v|\!)) \in u$ and $h(M_{q'}(\!|v|\!)) \in u$. Since $q > q'$, axiom A6 gives $h(\neg M_{q'}(\!|v|\!)) \in u$, which is a contradiction. ■

Finally we define the labelling function $\ell_\rho : \mathcal{U}[\rho] \to 2^{\mathcal{AP}}$ for any $u \in \mathcal{U}[\rho]$ as $\ell_\rho(u) = \{p \in \mathcal{AP} \mid p \in u\}$. We then have a model $\mathcal{M}_\rho = (\mathcal{U}[\rho], \to_\rho, \ell_\rho)$, and it is not difficult to prove that $\mathcal{M}_\rho$ is a WTS. Before we can prove the truth lemma, we need the following technical lemma.

Lemma A.4.7. *For any consistent formula $\varphi \in \mathcal{L}[\rho]$, if $[\mathcal{M}_\rho, u \models \varphi$ iff $h(\varphi) \in u]$, then*

$$\bigvee_{v \in [\![\varphi]\!]} h(\!|v|\!) \in u \quad \textit{iff} \quad h(\varphi) \in u.$$

Proof. Suppose $\bigvee_{v \in [\![\varphi]\!]} h(\!|v|\!) \in u$. Assume towards a contradiction that $h(\neg(\!|v|\!)) \in u$ for all $v \in [\![\varphi]\!]$. Then, since u is an ultrafilter, we must have $\bigwedge_{v \in [\![\varphi]\!]} h(\neg(\!|v|\!)) \in u$, which means that $\neg \bigvee_{v \in [\![\varphi]\!]} h(\!|v|\!) \in u$, which is a contradiction. Hence there exists some $v' \in [\![\varphi]\!]$ such that $h(\!|v'|\!) \in u$. If $\hat{\psi} \in v'$,

then $\vdash (\!|v'|\!) \to \psi$, so $\hat{\psi} \in u$ because u is an ultrafilter. Since $v' \in [\![\varphi]\!]$, we have by assumption that $h(\varphi) \in v'$, so we get $h(\varphi) \in u$.

Suppose $h(\varphi) \in u$, which by assumption means that $u \in [\![\varphi]\!]$, so

$$\vdash (\!|u|\!) \to \bigvee_{v \in [\![\varphi]\!]} (\!|v|\!).$$

Since u is an ultrafilter, we have $h((\!|u|\!)) \in u$, and hence $\bigvee_{v \in [\![\varphi]\!]} h((\!|v|\!)) \in u$. ■

We are now in a position to state and prove the truth lemma, which says that an ultrafilter satisfies a formula in our model if and only if that formula is included in the ultrafilter.

Lemma A.4.8 (Truth lemma)**.** *If $\rho \in \mathcal{L}$ is a consistent formula, then for all $\varphi \in \mathcal{L}[\rho]$ and $u \in \mathcal{U}[\rho]$ we have*

$$\mathcal{M}_\rho, u \models \varphi \quad \textit{iff} \quad h(\varphi) \in u.$$

Proof. The proof is by induction on the structure of φ. The Boolean cases are trivial. For the case $\varphi = L_r \psi$, we proceed as follows.

($\implies$) Assume $\mathcal{M}_\rho, u \models L_r \psi$, meaning that $\theta^-(u)([\![\psi]\!]) \geq r$. It can not be the case that $\theta(u)([\![\psi]\!]) = \emptyset$, because otherwise $\theta^-(u)([\![\psi]\!]) = -\infty$, and we have assumed $\theta^-(u)([\![\psi]\!]) \geq r$. It also can not be the case that $[\![\psi]\!] = \emptyset$, because otherwise $\theta(u)([\![\psi]\!]) = \emptyset$. We can partition all the ultrafilters $v \in [\![\psi]\!]$ as follows. Let $E = \{v \in [\![\psi]\!] \mid L(u, v) = \emptyset\}$ and $N = \{v \in [\![\psi]\!] \mid L(u, v) \neq \emptyset\}$. We then get that $E \cap N = \emptyset$, $E \cup N = [\![\psi]\!]$, $h(\neg L_0 (\!|v|\!)) \in u$ for all $v \in E$, and $h(L_r (\!|v|\!)) \in u$ for all $v \in N$. Because u is an ultrafilter, we then have

$$h\left(\bigwedge_{v \in E} \neg L_0 (\!|v|\!) \wedge \bigwedge_{v \in N} L_r (\!|v|\!)\right) \in u.$$

By axiom A3, this implies

$$h\left(\bigwedge_{v \in E} \neg L_0 (\!|v|\!) \wedge L_r \bigvee_{v \in N} (\!|v|\!)\right) \in u.$$

Then axiom A5 gives

$$h\left(L_r \bigvee_{v \in [\![\psi]\!]} (\!|v|\!)\right) \in u.$$

By the induction hypothesis, T2, and Lemma A.4.7, we then get $h(L_r \psi) \in u$.

($\impliedby$) Let $h(L_r \psi) \in u$. It follows from A1, A2, and R2 that ψ is consistent. Hence, by the induction hypothesis, $[\![\psi]\!]$ is non-empty. We first show that $\theta(u)([\![\psi]\!]) \neq \emptyset$. Assume therefore towards a contradiction that $\theta(u)([\![\psi]\!]) = \emptyset$. Then for all $v \in [\![\psi]\!]$, we must have that case 3 holds, and hence $L(u, v) = \emptyset$,

meaning $h(\neg L_r(\!|v|\!)) \in u$ for all $v \in [\![\psi]\!]$. Since there are finitely many $v \in [\![\psi]\!]$, we can enumerate them as $v_1, v_2, \ldots, v_n$. Then, since u is an ultrafilter, we have

$$h\left(\neg L_r(\!|v_1|\!) \wedge \neg L_r(\!|v_2|\!) \wedge \cdots \wedge \neg L_r(\!|v_n|\!)\right) \in u.$$

By De Morgan's law, this is equivalent to

$$h\left(\neg(L_r(\!|v_1|\!) \vee L_r(\!|v_2|\!) \vee \cdots \vee L_r(\!|v_n|\!))\right) \in u.$$

The contrapositive of axiom A4 then gives that

$$h\left(\neg L_r((\!|v_1|\!) \vee (\!|v_2|\!) \vee \cdots \vee (\!|v_n|\!))\right) \in u,$$

and by the induction hypothesis, T2, and Lemma A.4.7, this is equivalent to $\neg h(L_r\psi) \in u$, which is a contradiction.

Now assume towards a contradiction that $\theta^-(u)([\![\psi]\!]) < r$. Then there exists some $v \in [\![\psi]\!]$ such that $\theta^-(u)(\{v\}) < r$ and case 1 or case 2 holds. In either case we have $\max L(u, v) < r$ and hence there exists some $q \in Q_\rho$ such that $h(L_q(\!|v|\!)) \in u$, which implies $h(L_0(\!|v|\!)) \in u$ by axiom A2. By the induction hypothesis, $h(\psi) \in v$, which means that $\vdash (\!|v|\!) \to \psi$. rule R1 then gives $h(L_r(\!|v|\!)) \in u$, but this is a contradiction since $\max L(u, v) < r$.

The M_r case is similar, using axiom A7 instead of A2 to derive $h(L_0\psi) \in u$. ■

Having established the truth lemma, we can now show that any consistent formula is satisfied by some finite model.

Theorem A.4.9 (Finite model property)**.** *For any consistent formula $\varphi \in \mathcal{L}$, there exists a finite WTS $\mathcal{M} = (S, \to, \ell)$ and a state $s \in S$ such that $\mathcal{M}, s \models \varphi$.*

Proof. Since $\varphi \in \mathcal{L}$ is consistent, $h(\varphi) \neq h(\bot)$, and since $\mathcal{L}_{\leftrightarrow}[\rho]$ is finite, there must exist an ultrafilter $u \in \mathcal{U}[\rho]$ such that $h(\varphi) \in u$. By the truth lemma, this means that $\mathcal{M}_\varphi, u \models \varphi$, and by construction, $\mathcal{M}_\varphi$ is a finite model. ■

We are now able to state and prove our main result, namely that our axiomatisation is complete.

Theorem A.4.10 (Completeness)**.** *For any formula $\varphi \in \mathcal{L}$, it holds that*

$$\models \varphi \quad \textit{implies} \quad \vdash \varphi.$$

Proof.

$$\models \varphi \quad \text{implies} \quad \vdash \varphi$$

is equivalent to

$$\not\vdash \varphi \quad \text{implies} \quad \not\models \varphi,$$

which is equivalent to

the consistency of $\neg\varphi$ implies the existence of a model for $\neg\varphi$,

and this is guaranteed by the finite model property. ■

We have thus established completeness for our logic. There is also a stronger notion of completeness, often called strong completeness, which asserts that $\Phi \models \varphi$ implies $\Phi \vdash \varphi$ for any set of formulas $\Phi \subseteq \mathcal{L}$. Completeness is a special case of strong completeness where $\Phi = \emptyset$. In the case of compact logics, strong completeness follows directly from completeness. However, our logic is non-compact.

Theorem A.4.11. *Our logic is non-compact, meaning that there exists an infinite set $\Phi \subseteq \mathcal{L}$ such that each finite subset of Φ admits a model, but Φ does not.*

Proof. Consider the set

$$\Phi = \{L_q\varphi \mid q < r\} \cup \{\neg L_r\varphi\}.$$

For any finite subset of Φ, it is easy to construct a model. However, if $\mathcal{M}, s \models L_q\varphi$ for all $q < r$ where $q, r \in \mathbb{Q}_{\geq 0}$, then by the Archimedean property of the rationals, we also have $\mathcal{M}, s \models L_r\varphi$. Hence there can be no model for Φ. ■

A.5 Model Checking and Satisfiability

We now turn our attention to decision problems related to our logic. First we consider the model checking problem, which asks us to decide whether $\mathcal{M}, s \models \varphi$ for a given model $\mathcal{M}$, state s, and formula φ. We will develop a polynomial time algorithm for this problem by adapting the classical model checking algorithm of Clarke et al. [8] to our setting. In what follows, we will assume that all models have a finite state space, and that each state has finitely many outgoing transitions. Furthermore, we will assume that the set $\mathcal{AP}$ of atomic propositions is finite.

Given a formula φ and a model $\mathcal{M} = (S, \to, \ell)$, we construct a function $F_\varphi : S \to 2^{\mathcal{L}}$ which assigns a set of formulas to each state. Intuitively, $F_\varphi(s)$ will be the set of subformulas of φ that are true in s.

In order to do this, we first introduce the following terminology. The *closure* of a formula φ, denoted $\mathrm{cl}(\varphi)$, is given by

$$\mathrm{cl}(\varphi) = \begin{cases} \{p\} & \text{if } \varphi = p \\ \mathrm{cl}(\varphi') \cup \{\varphi\} & \text{if } \varphi = \neg\varphi', \varphi = L_r\varphi', \text{ or } \varphi = M_r\varphi' \\ \mathrm{cl}(\varphi_1) \cup \mathrm{cl}(\varphi_2) \cup \{\varphi\} & \text{if } \varphi = \varphi_1 \wedge \varphi_2 \end{cases}$$

Definition A.5.1. A formula φ' is said to be a *subformula* of φ if $\varphi' \in \mathrm{cl}(\varphi)$. φ' is said to be a *proper* subformula of φ if it is a subformula and $\varphi' \neq \varphi$. ▲

Definition A.5.2. For a formula φ, we define the length of φ as follows.

- φ has length 1 if it has no proper subformulas.

- φ has length i if its longest proper subformula has length $i-1$.

We will denote the length of φ by $\text{len}(\varphi)$. ▲

We can now construct the function F_φ by means of Algorithm A.5.1.

Lemma A.5.3. *Let $\mathcal{M} = (S, \to, \ell)$ be a model, $s \in S$ a state, and φ a formula. For any subformula φ' of φ it holds that*

$$\mathcal{M}, s \models \varphi' \quad \textit{if and only if} \quad \varphi' \in F_\varphi(s).$$

Proof. We will prove this by structural induction on φ'.

($\varphi' = p$):

$$\begin{aligned} \mathcal{M}, s \models p \text{ iff } & p \in \ell(s) && \text{(Definition A.3.2)} \\ \text{iff } & p \in F_\varphi(s) && \text{(Algorithm A.5.1).} \end{aligned}$$

($\varphi' = \neg\psi$):

$$\begin{aligned} \mathcal{M}, s \models \neg\psi \text{ iff } & \mathcal{M}, s \not\models \psi && \text{(Definition A.3.2)} \\ \text{iff } & \psi \notin F_\varphi(s) && \text{(ind. hyp.)} \\ \text{iff } & \neg\psi \in F_\varphi(s) && \text{(Algorithm A.5.1).} \end{aligned}$$

($\varphi' = \psi_1 \wedge \psi_2$):

$$\begin{aligned} \mathcal{M}, s \models \psi_1 \wedge \psi_2 \text{ iff } & \mathcal{M}, s \models \psi_1 \text{ and } \mathcal{M}, s \models \psi_2 && \text{(Definition A.3.2)} \\ \text{iff } & \psi_1 \in F_\varphi(s) \text{ and } \psi_2 \in F_\varphi(s) && \text{(ind. hyp.)} \\ \text{iff } & \psi_1 \wedge \psi_2 \in F_\varphi(s) && \text{(Algorithm A.5.1).} \end{aligned}$$

($\varphi' = L_r\psi$):

$$\begin{aligned} \mathcal{M}, s \models L_r\psi \text{ iff } & \theta^-(s)([\![\psi]\!]) \geq r && \text{(Definition A.3.2)} \\ \text{iff } & \theta^-(s)(S_\psi) \geq r && \text{(ind. hyp.)} \\ \text{iff } & S_\psi \neq \emptyset \text{ and} \\ & \quad \min\{r' \mid \exists s' \in S_\psi . s \xrightarrow{r'} s'\} \geq r \\ \text{iff } & L_r\psi \in F_\varphi(s) && \text{(Algorithm A.5.1).} \end{aligned}$$

($\varphi' = M_r\psi$):

$$\begin{aligned} \mathcal{M}, s \models M_r\psi \text{ iff } & \theta^+(s)([\![\psi]\!]) \leq r && \text{(Definition A.3.2)} \\ \text{iff } & \theta^+(s)(S_\psi) \leq r && \text{(ind. hyp.)} \\ \text{iff } & S_\psi \neq \emptyset \text{ and} \\ & \quad \max\{r' \mid \exists s' \in S_\psi . s \xrightarrow{r'} s'\} \leq r \\ \text{iff } & M_r\psi \in F_\varphi(s) && \text{(Algorithm A.5.1).} \end{aligned}$$

■

```
1  Let F_φ(s) = ∅ for all s ∈ S ;
2  Let Φ_i be the set of all subformulas of φ of length i ;
3  for i = 1, ..., len(φ) do
4      for φ' ∈ Φ_i do
5          for s ∈ S do
6              if φ' = p then
7                  if p ∈ ℓ(s) then
8                      F_φ(s) := F_φ(s) ∪ {φ'} ;
9                  end
10             end
11             if φ' = ¬ψ then
12                 if ψ ∉ F_φ(s) then
13                     F_φ(s) := F_φ(s) ∪ {φ'} ;
14                 end
15             end
16             if φ' = ψ_1 ∧ ψ_2 then
17                 if ψ_1 ∈ F_φ(s) and ψ_2 ∈ F_φ(s) then
18                     F_φ(s) := F_φ(s) ∪ {φ'} ;
19                 end
20             end
21             if φ' = L_r ψ then
22                 Let S_ψ = {s' ∈ S | ψ ∈ F_φ(s')} ;
23                 if S_ψ ≠ ∅ and min{r' | s --r'--> s' for some s' ∈ S_ψ} ≥ r then
24                     F_φ(s) := F_φ(s) ∪ {φ'} ;
25                 end
26             end
27             if φ' = M_r ψ then
28                 Let S_ψ = {s' ∈ S | ψ ∈ F_φ(s')} ;
29                 if S_ψ ≠ ∅ and max{r' | s --r'--> s' for some s' ∈ S_ψ} ≤ r then
30                     F_φ(s) := F_φ(s) ∪ {φ'} ;
31                 end
32             end
33         end
34     end
35 end
36 return F_φ ;
```

Algorithm A.5.1: Constructing the function F_φ for a given formula φ.

We can now prove that, given a model $\mathcal{M} = (S, \to, \ell)$ and formula φ, the model checking problem is decidable in polynomial time.

Theorem A.5.4. *The model checking problem for our logic is decidable in time*

$$\mathcal{O}(|\mathrm{cl}(\varphi)| \cdot |S| \cdot (n \cdot \log n + |\mathcal{AP}| + |\mathrm{cl}(\varphi)|)),$$

where n is the degree of $\mathcal{M}$, i.e. the maximum number of outgoing transitions from a state in S.

Proof. It follows from Lemma A.5.3 that the model checking problem is decidable by constructing the function F_φ and checking whether $\varphi \in F_\varphi(s)$. It therefore remains to argue that Algorithm A.5.1 runs in the claimed time.

The first two loops in Algorithm A.5.1 at line 3 and 4 iterate over all elements of $\mathrm{cl}(\varphi)$, and the third loop iterates over all elements of S. Inside the third loop, the algorithm enters one of the if-statements depending on the structure of the current subformula. If it enters the first if-statement at line 7, then we must search $\ell(s)$ for p. This takes at most time $|\mathcal{AP}|$. If it enters the second or third if-statement at line 11 and 16, then we must search $F_\varphi(s)$ for one of the formulas. This takes at most time $|\mathrm{cl}(\varphi)|$, since the function F_φ can only label states with subformulas of φ. Lastly, if the algorithm enters the fourth of fifth if-statement at line 21 and 27, then we must find the corresponding minimum and maximum value. This can be done using e.g. mergesort in time $n \cdot \log n$.

Together, this analysis of Algorithm A.5.1 gives the claimed run-time. ■

Next we will consider the satisfiability problem, which asks us to decide whether a given formula φ has a model or not. The finite model property gives us a way of deciding this problem. An algorithm would be to enumerate all finite models and all theorems derivable from the axioms, which can be done since there are countably many of each of these. If φ is satisfiable, it has a model, and by the finite model property, it has a finite one. So we can check one by one whether a finite model satisfies φ by using the model checking algorithm described previously. On the other hand, if φ is not satisfiable, then $\neg\varphi$ is a theorem, so we can search through all theorems to see whether $\neg\varphi$ is one of them. Since φ is either satisfiable or its negation is a theorem, one of these two algorithms must eventually halt. By running these two algorithms in parallel, we have shown that the problem of deciding satisfiability for a given formula is decidable.

In what follows we do more: We propose an algorithm that constructs a tableau syntactically from a given formula. By inspecting this tableau, we can decide whether or not the formula is satisfiable, and if it is satisfiable, we can construct a model for the formula from the tableau.

As in the previous section, we impose an order on formulas given by $\varphi \leq \psi$ if and only if $\models \varphi \to \psi$. Given a finite set of formulas $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$,

$$(\wedge)\ \frac{\langle \Gamma \cup \{\varphi \wedge \psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\varphi, \psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\neg\wedge)\ \frac{\langle \Gamma \cup \{\neg(\varphi \wedge \psi)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\neg\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad \langle \Gamma \cup \{\neg\psi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\neg\neg)\ \frac{\langle \Gamma \cup \{\neg\neg\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \Gamma \cup \{\varphi\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

$$(\text{mod})\ \frac{\langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1 \rangle \quad \cdots \quad \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k \rangle}$$

if $N^i \in \{L, M\}$ for all $1 \leq i \leq n$, $O^j \in \{L, M\}$ for all $1 \leq j \leq n'$, and no formula in Γ is of the form $N_r\varphi$ or $\neg N_r\varphi$ where $N \in \{L, M\}$.

Table A.5.1: Tableau rules.

we denote by $\min(\Gamma)$ the set of minimal elements of Γ, i.e.

$$\min(\Gamma) = \{\varphi_i \in \Gamma \mid \text{there is no } \varphi_j \text{ such that } \varphi_j \leq \varphi_i\},$$

and we let

$$\mathcal{L}(\Gamma) = \{\varphi_i \in \Gamma \mid \text{there is no } j < i \text{ such that } \models \varphi_j \leftrightarrow \varphi_i\}.$$

Furthermore, we let $\uparrow_\Gamma(\varphi)$ be the upward closure of φ in Γ, i.e.

$$\uparrow_\Gamma(\varphi) = \{\varphi' \in \Gamma \mid \varphi \leq \varphi'\}.$$

A *tableau* is a tree with nodes of the form $\langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle$ that is constructed from the rules of Table A.5.1, where the (mod) rule may only be used when no other rule can be used. For each node $\langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle$, Γ is a set of formulas, and $\mathcal{I}^L$ and $\mathcal{I}^M$ are intervals of the form $\wr a, b \int$ where $a \in \mathbb{R}_{\geq 0} \cup \{-\infty\}$, $b \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, $\wr \in \{[, (\}$, and $\int \in \{],)\}$, subject to the constraint that $\wr = ($ if $a = -\infty$ and $\int =)$ if $b = \infty$. We will say that an interval $\wr a, b \int$ is *consistent* if $a < b$ or $a = b$ and the interval is closed.

For the rule (mod), the objects ψ_i, $\mathcal{I}^L_i$ and $\mathcal{I}^M_i$ in the conclusion are constructed as follows. The ψ_i are given by

$$\{\psi_1, \ldots, \psi_k\} = \min(\mathcal{L}(\{\varphi_1, \ldots, \varphi_n\})).$$

Let $\Gamma' = \{\varphi_1, \ldots, \varphi_n\}$ and

$$\mathbb{L}_i^+ = \{r \mid L_r\varphi_j = N_{r_j}^j\varphi_j \text{ for some } j \text{ and } \varphi_j \in \uparrow_{\Gamma'}(\psi_i)\}$$

$$\mathbb{M}_i^+ = \{r \mid M_r\varphi_j = N_{r_j}^j\varphi_j \text{ for some } j \text{ and } \varphi_j \in \uparrow_{\Gamma'}(\psi_i)\}$$

as well as

$$\mathbb{L}_i^- = \{r \mid L_r\varphi'_j = O_{r_j}^j\varphi'_j \text{ for some } j \text{ and } \models \psi_i \to \varphi'_j\}$$

$$\mathbb{M}_i^- = \{r \mid M_r\varphi'_j = O_{r_j}^j\varphi'_j \text{ for some } j \text{ and } \models \psi_i \to \varphi'_j\}.$$

Then the intervals $\mathcal{I}_i^L$ and $\mathcal{I}_i^M$ are given by

$$\mathcal{I}_i^L = \begin{cases} [\max \mathbb{L}_i^+, \min \mathbb{L}_i^-) & \text{if } \mathbb{L}_i^+ \neq \emptyset \text{ and } \mathbb{L}_i^- \neq \emptyset \\ [0, \min \mathbb{L}_i^-) & \text{if } \mathbb{L}_i^+ = \emptyset \text{ and } \mathbb{L}_i^- \neq \emptyset \\ [\max \mathbb{L}_i^+, \infty) & \text{if } \mathbb{L}_i^+ \neq \emptyset \text{ and } \mathbb{L}_i^- = \emptyset \\ [0, \infty) & \text{if } \mathbb{L}_i^+ = \emptyset \text{ and } \mathbb{L}_i^- = \emptyset \end{cases}$$

$$\mathcal{I}_i^M = \begin{cases} (\max \mathbb{M}_i^-, \min \mathbb{M}_i^+] & \text{if } \mathbb{M}_i^- \neq \emptyset \text{ and } \mathbb{M}_i^+ \neq \emptyset \\ [0, \min \mathbb{M}_i^+] & \text{if } \mathbb{M}_i^- = \emptyset \text{ and } \mathbb{M}_i^+ \neq \emptyset \\ (\max \mathbb{M}_i^-, \infty) & \text{if } \mathbb{M}_i^- \neq \emptyset \text{ and } \mathbb{M}_i^+ = \emptyset \\ [0, \infty) & \text{if } \mathbb{M}_i^- = \emptyset \text{ and } \mathbb{M}_i^+ = \emptyset \end{cases}$$

Informally, one should think of a node $m = \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle$ as satisfying all the formulas in Γ. Moreover, the (mod)-rule signifies a state transition, where the new states are given by the nodes in the conclusion, and any transition to m must have a minimum weight that lies in the interval $\mathcal{I}^L$, and a maximum weight that lies in the interval $\mathcal{I}^M$.

Example A.5.5. We now illustrate the use of the (mod) rule through an example. Consider the node

$$m = \langle \{p_1, p_2, L_2p_1, L_4(p_1 \wedge p_2), L_0p_3, \neg L_5p_2, \neg M_6p_3\}, \mathcal{I}^L, \mathcal{I}^M \rangle.$$

We group the formulas as

$$\Gamma = \{p_1, p_2\}, \Gamma' = \{L_2p_1, L_4(p_1 \wedge p_2), L_0p_3\}, \text{ and } \Gamma'' = \{\neg L_5p_2, \neg M_6p_3\},$$

so that $m = \langle \Gamma \cup \Gamma' \cup \Gamma'', \mathcal{I}^L, \mathcal{I}^M \rangle$. Since Γ only includes literals, it is clear that we can use no other rules, so we are allowed to use (mod) on m.

We see that $\models (p_1 \wedge p_2) \to p_1$, and hence $\{\psi_1, \psi_2\} = \{p_1 \wedge p_2, p_3\}$, so there are two children of m. For the first child, we find

$$\mathbb{L}_1^+ = \{2, 4\} \qquad \mathbb{M}_1^+ = \emptyset$$
$$\mathbb{L}_1^- = \{5\} \qquad \mathbb{M}_1^- = \emptyset,$$

and for the second child we find

$$\mathbb{L}_2^+ = \{0\} \qquad \mathbb{M}_2^+ = \emptyset$$
$$\mathbb{L}_2^- = \emptyset \qquad \mathbb{M}_2^- = \{6\}.$$

Hence the intervals become

$$\mathcal{I}_1^L = [4,5) \qquad \mathcal{I}_1^M = [0,\infty)$$
$$\mathcal{I}_2^L = [0,\infty) \qquad \mathcal{I}_2^M = (6,\infty),$$

and our application of the rule becomes

$$\text{(mod)}\ \frac{\langle \{p_1, p_2, L_2 p_1, L_4(p_1 \wedge p_2), L_0 p_3, \neg L_5 p_2, \neg M_6 p_3\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{\langle \{p_1 \wedge p_2\}, [4,5), [0,\infty) \rangle \quad \langle \{p_3\}, [0,\infty)(6,\infty) \rangle} \qquad \blacklozenge$$

Given a formula φ, we will say that a tableau $\mathcal{T}$ is a *tableau for* φ if $\langle \{\varphi\}, [0,0], [0,0] \rangle$ is the root of $\mathcal{T}$. A node m in a tableau is called a *modal node* if the (mod)-rule was applied to m. We will say that a node is a *terminal node* if it is either a modal node or a leaf node.

Definition A.5.6. A node $m = \langle \Gamma, \wr_1 a, b \int_1, \wr_2 c, d \int_2 \rangle$ is *consistent* if

- for any $p \in \mathcal{AP}$ we do not have both $p \in \Gamma$ and $\neg p \in \Gamma$,
- $\wr_1 a, b \int_1$ and $\wr_2 c, d \int_2$ are consistent, and
- either $a < d$ or $a = d$, $\wr_1 = [$, and $\int_2 =]$. ▲

Definition A.5.7. A tableau $\mathcal{T}$ is *successful* if there exists a subtree $\mathcal{T}'$ of $\mathcal{T}$ such that

- every leaf in $\mathcal{T}'$ is also a leaf in $\mathcal{T}$,
- if a modal node m is included in $\mathcal{T}'$, then every child of m is also included in $\mathcal{T}'$, and
- every terminal node in $\mathcal{T}'$ is consistent. ▲

Given a successful tableau $\mathcal{T}$, we construct the WTS $\mathcal{M}(\mathcal{T})$ with state $s_{\mathcal{T}}$ using Algorithm A.5.2.

Lemma A.5.8. *If $\mathcal{T}$ is a successful tableau for φ, then $\mathcal{M}(\mathcal{T}), s_{\mathcal{T}} \models \varphi$.*

Proof. Let Y be the set of all pairs (s, m) that are added to the stack X by Algorithm A.5.2 at some point during the construction of $\mathcal{M}(\mathcal{T})$. We wish to prove that for any $(s, \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle) \in Y$ we have $\mathcal{M}(\mathcal{T}), s \models \Gamma$, where we write $\mathcal{M}(\mathcal{T}), s \models \Gamma$ to mean $\mathcal{M}(\mathcal{T}), s \models \varphi$ for all $\varphi \in \Gamma$. Note that if we can prove this, then it follows that $\mathcal{M}(\mathcal{T}), s_{\mathcal{T}} \models \varphi$ since $(s_{\mathcal{T}}, \langle \{\varphi\}, \emptyset, (0,0) \rangle) \in Y$.

Let $\mathcal{T}'$ be a witness for the fact that $\mathcal{T}$ is successful ;
$S := \{s_{\mathcal{T}}\}, \rightarrow := \emptyset, \ell := \emptyset$;
Let X be a stack and $X := \emptyset$;
$X.push((s_{\mathcal{T}}, r))$ where r is the root of $\mathcal{T}'$;
while $X \neq \emptyset$ **do**
    $(s, m) := X.pop$;
    Let $m = \langle \Gamma, \Delta, (a, b) \rangle$;
    if *m is not a terminal node* **then**
        Let m' be the left-most child of m in $\mathcal{T}'$;
        $X.push((s, m'))$;
    end
    if *m is a leaf node* **then**
        $\ell := \ell \cup \{(s, p) \mid p \in \mathcal{AP}$ and $p \in \Gamma\}$;
    end
    if *m is a modal node* **then**
        $\ell := \ell \cup \{(s, p) \mid p \in \mathcal{AP}$ and $p \in \Gamma\}$;
        Let $m_1 = \langle \Gamma_1, \mathcal{I}_1^L, \mathcal{I}_1^M \rangle, \ldots, m_n = \langle \Gamma_n, \mathcal{I}_n^L, \mathcal{I}_n^M \rangle$ be the children of m in $\mathcal{T}'$;
        for $i = 1, \ldots, n$ **do**
            Let $\mathcal{I}_i^L = \lbag a_i, b_i \rbag$ and $\mathcal{I}_i^M = \lbag c_i, d_i \rbag$;
            $x_i := a_i$;
            $y_i := \begin{cases} \max\{a_i, \frac{d_i - c_i}{2} + c_i\} & \text{if } d_i \neq \infty \\ \max\{a_i, c_i + 1\} & \text{if } d_i = \infty \end{cases}$;
            $S := S \cup \{s_i\}$;
            $\rightarrow := \rightarrow \cup \{(s, x_i, s_i), (s, y_i, s_i)\}$;
            $X.push((s_i, m_i))$;
        end
    end
end
$\mathcal{M}(\mathcal{T}) := (S, \rightarrow, \ell)$;
return $(\mathcal{M}(\mathcal{T}), s_{\mathcal{T}})$;

Algorithm A.5.2: Constructing the model $\mathcal{M}(\mathcal{T})$ for a successful tableau $\mathcal{T}$.

Let (s, m) be an arbitrary element of Y and let l be the length of the longest path from m to a leaf. We will prove, by induction on l, that $\mathcal{M}(\mathcal{T}), s \models \Gamma$ where $m = \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle$.

$l = 0$: In this case, m is a leaf. Hence Γ only contains literals, and by construction we have $p \in \ell(s)$ if and only if $p \in \Gamma$. Since m is consistent, we thus get $\mathcal{M}(\mathcal{T}), s \models \Gamma$.

$l > 0$: In this case we consider the different rules that may be applied to m.

($\wedge$) We have

$$(\wedge)\ \frac{m = \langle \Gamma \cup \{\varphi_1 \wedge \varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m' = \langle \Gamma \cup \{\varphi_1, \varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

By induction hypothesis we get $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\varphi_1, \varphi_2\}$. This implies that $\mathcal{M}(\mathcal{T}), s \models \varphi_1$ and $\mathcal{M}(\mathcal{T}), s \models \varphi_2$, so $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\varphi_1 \wedge \varphi_2\}$.

($\neg\wedge$) We have

$$(\neg\wedge)\ \frac{m = \langle \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \Gamma \cup \{\neg\varphi_1\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad m_2 = \langle \Gamma \cup \{\neg\varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

We have three cases to consider; either m_1 is included in $\mathcal{T}'$, m_2 is included in $\mathcal{T}'$, or both m_1 and m_2 are included in $\mathcal{T}'$. If m_1 is included in $\mathcal{T}'$ we get, by the induction hypothesis, that $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\neg\varphi_1\}$ implying that $\mathcal{M}(\mathcal{T}), s \not\models \varphi_1$. If m_2 is included in $\mathcal{T}'$ we get, by the induction hypothesis, that $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\neg\varphi_2\}$ implying that $\mathcal{M}(\mathcal{T}), s \not\models \varphi_2$. In either case we get that $\mathcal{M}(\mathcal{T}), s \not\models \varphi_1 \wedge \varphi_2$ and $\mathcal{M}(\mathcal{T}), s \models \Gamma$, and therefore $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}$. The last case follows trivially from the preceding arguments.

($\neg\neg$) We have

$$(\neg\neg)\ \frac{m = \langle \Gamma \cup \{\neg\neg\varphi'\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m' = \langle \Gamma \cup \{\varphi'\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

By induction hypothesis we know that $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\varphi'\}$, and hence $\mathcal{M}(\mathcal{T}), s \models \Gamma \cup \{\neg\neg\varphi'\}$.

(mod) We have

$$(\text{mod})\ \frac{m = \langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1 \rangle \quad \cdots \quad m_k = \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k \rangle}$$

Γ must consist only of literals, because otherwise the (mod) rule could not be used. As in the case for $l = 0$, we then get $\mathcal{M}(\mathcal{T}), s \models \Gamma$ since m is consistent. Let $\Psi = \{\psi_1, \ldots, \psi_k\}$, and for any $1 \leq j \leq k$, let $\mathcal{I}^L_j =$

$\wr a_j, b_j \int$ and $\mathcal{I}_j^M = \wr c_j, d_k \int$. By the induction hypothesis, we know that $\mathcal{M}(\mathcal{T}), s_j \models \psi_j$ for all $j \in \{1, \ldots, k\}$, and, by construction, s_j is the only successor of s that satisfies ψ_j. Now consider a formula $N^i_{r_i}\varphi_i$. There must exist a subset $\Psi_{\varphi_i} \subseteq \Psi$ such that $\theta(s)(\llbracket\varphi_i\rrbracket) = \theta(s)\left(\bigcup_{\psi' \in \Psi_{\varphi_i}} \llbracket\psi'\rrbracket\right)$. We first consider the case where $N^i = L$. Because Ψ_{φ_i} is finite, there exists $\psi'_j \in \Psi_{\varphi_i}$ such that $\theta^-(s)(\llbracket\varphi_i\rrbracket) = \theta^-(s)\left(\llbracket\psi'_j\rrbracket\right)$, implying the existence of $\psi_j \in \Psi$ such that $\theta^-(s)(\llbracket\varphi_i\rrbracket) = \theta^-(s)(\llbracket\psi_j\rrbracket) = a_j$. We must have $a_j \geq r_i$ implying $\theta^-(s)(\llbracket\varphi_i\rrbracket) \geq r_i$, and thus $\mathcal{M}(\mathcal{T}), s \models L_{r_i}\varphi_i$. In the case where $N^i = M$ we can, similarly to the previous case, find $\psi_j \in \Psi$ such that $\theta^+(s)(\llbracket\varphi_i\rrbracket) = \theta^+(s)(\llbracket\psi_j\rrbracket)$, and we know that $d_i \neq \infty$ implying

$$\theta^+(s)(\llbracket\psi_j\rrbracket) = \max\left\{a_j, \frac{d_j - c_j}{2} + c_j\right\} \leq d_j \leq r_i.$$

Therefore, $\theta^+(s)(\llbracket\varphi\rrbracket) \leq r_i$ and thus $\mathcal{M}(\mathcal{T}), s \models M_{r_i}\varphi_i$.

Lastly we consider a formula $\neg O^i_{r'_i}\varphi'_i$. If there is no $\psi_j \in \Psi$ such that $\models \psi_j \to \varphi'_i$, then, by the construction of $\mathcal{M}(\mathcal{T})$, there is no successor s' of s such that $\mathcal{M}(\mathcal{T}), s \models \varphi'_i$. Therefore, $\theta^-(s)(\llbracket\varphi'_i\rrbracket) = \infty$ and $\theta^+(s)(\llbracket\varphi'_i\rrbracket) = -\infty$, and thus $\mathcal{M}(\mathcal{T}), s \models \neg O^i_{r'_i}\varphi'_i$ is trivially satisfied for $O^i \in \{L, M\}$. Suppose $\models \psi'_j \to \varphi'_i$ for some $\psi'_j \in \Psi$. We first consider the case where $O^i = L$. There must exist $\psi_j \in \Psi$ such that $\theta^-(s)(\llbracket\varphi'_i\rrbracket) = \theta^-(s)(\llbracket\psi_j\rrbracket) = a_j$. By the assumption that $\mathcal{T}$ is successful, we must have that m_j is consistent. Therefore, $a_j < b_j \leq r_{i'}$ implying $\theta^-(s)(\llbracket\varphi'_i\rrbracket) < r'_i$, and thus $\mathcal{M}(\mathcal{T}), s \models \neg L_{r'_i}\varphi'_i$. In the case where $O^i = M$ we must be able to find $\psi_j \in \Psi$ such that $\theta^+(s)(\llbracket\varphi'_i\rrbracket) = \theta^+(s)(\llbracket\psi_j\rrbracket)$. We have to consider $d_j = \infty$ and $d_j \neq \infty$ separately. If $d_j = \infty$ we have

$$\theta^+(s)(\llbracket\psi_j\rrbracket) = \max\{a_j, c_j + 1\} > c_j \geq r'_i.$$

If $d_j \neq \infty$ we have

$$\theta^+(s)(\llbracket\psi_j\rrbracket) = \max\left\{a_j, \frac{d_j - c_j}{2} + c_j\right\} > c_j \geq r'_i.$$

In either case we have that $\theta(s)(\llbracket\varphi'_i\rrbracket) > r'_i$, so $\mathcal{M}(\mathcal{T}), s \models \neg M_{r'_i}\varphi'_i$. ■

Lemma A.5.9. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be tableaux for φ. Then it holds that $\mathcal{T}_1$ is successful if and only if $\mathcal{T}_2$ is successful.*

Proof. Assume that $\mathcal{T}_1$ is a successful tableau. Let $\mathcal{T}'_1$ be a subtree of $\mathcal{T}_1$ which witnesses the fact that $\mathcal{T}_1$ is successful. If $\mathcal{T}'_1$ is also a subtree of $\mathcal{T}_2$, then we

are done. If not, let d be the smallest number such that $\mathcal{T}_1'$ differs at depth d from any subtree of $\mathcal{T}_2$ with the same root as $\mathcal{T}_2$. Note that we must have $d > 0$ because $\mathcal{T}_1'$ and $\mathcal{T}_2$ have the same root. Denote by $\mathcal{T}_1'|_n$ the restriction of $\mathcal{T}_1'$ to depth n. Then $\mathcal{T}_1'|_{d-1}$ is a subtree of $\mathcal{T}_2$.

Let X be the set of all terminal nodes that are in $\mathcal{T}_1'$ at depth d or below. Then every node in X is also a node in $\mathcal{T}_2$, and furthermore, every node in X is reachable in $\mathcal{T}_2$ from $\mathcal{T}_1'|_{d-1}$. Hence, if we extend $\mathcal{T}_1'|_{d-1}$ to include all paths leading to an element in X, then this extension is a subtree of $\mathcal{T}_2$ that witnesses the fact that $\mathcal{T}_2$ is successful. ■

Lemma A.5.10. *φ is satisfiable if and only if there exists a successful tableau for φ.*

Proof. ($\implies$) Assume φ is satisfiable, meaning that $\mathcal{M}, s \models \varphi$ for some $\mathcal{M} = (S, \to, \ell)$ and $s \in S$.

Let $\mathcal{T}$ be a tableau for φ, and note that such a tableau always exists by applying the tableau rules to $\langle \{\varphi\}, [0,0], [0,0] \rangle$. Now construct a marking $\mathfrak{M} \subseteq S \times \mathcal{T}$ as follows.

- $(s, r) \in \mathfrak{M}$ where r is the root of $\mathcal{T}$.
- If $(s', m) \in \mathfrak{M}$ and $(\wedge)$ or $(\neg\neg)$ was applied to m, add (s', m') to $\mathfrak{M}$, where m' is the child of m.
- If $(s', m) \in \mathfrak{M}$ and $(\neg\wedge)$ was applied to m, meaning that

$$(\neg\wedge)\ \frac{m = \langle \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \Gamma \cup \{\neg\varphi_1\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad m_2 = \langle \Gamma \cup \{\neg\varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

  then add (s', m_1) to $\mathfrak{M}$ if $s' \in [\![\neg\varphi_1]\!]$ and add (s', m_2) to $\mathfrak{M}$ if $s' \in [\![\neg\varphi_2]\!]$.

- If $(s', m) \in \mathfrak{M}$ and (mod) was applied to m, meaning that

$$(\text{mod})\ \frac{m = \langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1 \rangle \quad \cdots \quad m_k = \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k \rangle}$$

  then add (t', m_i) to $\mathfrak{M}$ if $t' \in [\![\psi_i]\!]$ and $s' \xrightarrow{r} t'$ for some $r \in \mathbb{R}_{\geq 0}$.

We will first argue that for any $(s', \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle) \in \mathfrak{M}$ we have $\mathcal{M}, s' \models \Gamma$, meaning $\mathcal{M}, s' \models \varphi'$ for all $\varphi' \in \Gamma$. We prove this by induction on the depth d of m.

$d = 0$: We have $(s', \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M \rangle) = (s, r) = (s, \langle \{\varphi\}, [0,0], [0,0] \rangle)$, and by assumption we get $\mathcal{M}, s \models \varphi$.

$d > 0$: We consider which rule was applied to the parent of m.

$(\wedge)$:

$$(\wedge)\ \frac{m' = \langle \Gamma \cup \{\varphi_1 \wedge \varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m = \langle \Gamma \cup \{\varphi_1, \varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

By induction hypothesis, we have $\mathcal{M}, s' \models \Gamma \cup \{\varphi_1 \wedge \varphi_2\}$, so $\mathcal{M}, s' \models \varphi_1$ and $\mathcal{M}, s' \models \varphi_2$, and hence $\mathcal{M}, s' \models \Gamma \cup \{\varphi_1, \varphi_2\}$.

($\neg\wedge$):

$$(\neg\wedge)\ \frac{m' = \langle \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \Gamma \cup \{\neg\varphi_1\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad m_2 = \langle \Gamma \cup \{\neg\varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

If $m = m_1$, then by the way $\mathfrak{M}$ was constructed we get $\mathcal{M}, s' \models \neg\varphi_1$, and hence by induction hypothesis, $\mathcal{M}, s' \models \Gamma \cup \{\neg\varphi_1\}$. Likewise we get $\mathcal{M}, s' \models \Gamma \cup \{\neg\varphi_2\}$ if $m = m_2$.

($\neg\neg$):

$$(\neg\neg)\ \frac{m' = \langle \Gamma \cup \{\neg\neg\varphi'\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m = \langle \Gamma \cup \{\varphi'\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

By induction hypothesis we have $\mathcal{M}, s' \models \Gamma \cup \{\neg\neg\varphi'\}$, which is equivalent to $\mathcal{M}, s' \models \Gamma \cup \{\varphi'\}$.

(mod):

$$(\text{mod})\ \frac{m' = \langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1 \rangle \quad \cdots \quad m_k = \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k \rangle}$$

We must have $m = m_i$ for some $1 \leq i \leq k$. By construction of $\mathfrak{M}$ we know that $\mathcal{M}, m_i \models \psi_i$.

Now let $\mathcal{T}'$ be the subtree of $\mathcal{T}$ consisting of those nodes m where there exists a state s' such that $(s', m) \in \mathfrak{M}$. We will now prove that $\mathcal{T}'$ satisfies the three conditions in Definition A.5.7.

For the first condition we prove the contrapositive: If m is not a leaf in $\mathcal{T}$, then it is not a leaf in $\mathcal{T}'$. Hence we assume that m is not a leaf in $\mathcal{T}$. If m is not a node in $\mathcal{T}'$, then it is also not a leaf node in $\mathcal{T}'$. If m is a node in $\mathcal{T}'$, then there must exist some state s' such that $(s', m) \in \mathfrak{M}$. We now consider which rule was applied to m in $\mathcal{T}$.

($\wedge$) or ($\neg\neg$) In these cases, m has a child m' in $\mathcal{T}$, and by construction of $\mathfrak{M}$, we get $(s', m') \in \mathfrak{M}$, so m' is a child of m in $\mathcal{T}'$.

($\neg\wedge$)

$$(\neg\wedge)\ \frac{m = \langle \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}, \mathcal{I}^L, \mathcal{I}^M \rangle}{m_1 = \langle \Gamma \cup \{\neg\varphi_1\}, \mathcal{I}^L, \mathcal{I}^M \rangle \quad m_2 = \langle \Gamma \cup \{\neg\varphi_2\}, \mathcal{I}^L, \mathcal{I}^M \rangle}$$

We know that $\mathcal{M}, s' \models \Gamma \cup \{\neg(\varphi_1 \wedge \varphi_2)\}$, so we must have $\mathcal{M}, s' \models \neg\varphi_1$ or $\mathcal{M}, s' \models \neg\varphi_2$. By construction of $\mathfrak{M}$, this means that $(s', m_1) \in \mathfrak{M}$ or $(s', m_2) \in \mathfrak{M}$, and hence m_1 or m_2 must be a child of m in $\mathcal{T}'$.

(mod)

$$\text{(mod)}\ \frac{m = \langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M\rangle}{m_1 = \langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1\rangle \quad \cdots \quad m_k = \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k\rangle}$$

For each m_i there must exist some j such that $N^j_{r_j}\varphi_j = N^j_{r_j}\psi_i$. Then we know that $\mathcal{M}, s' \models N^j_{r_j}\psi_i$, and hence

$$\theta^-(s')([\![\psi_i]\!]) \geq r_j \quad \text{or} \quad \theta^+(s')([\![\psi_i]\!]) \leq r_j.$$

In either case there must exist some $t' \in [\![\psi_i]\!]$ such that $s' \xrightarrow{r} t'$ for some r. Hence $(t', m_i) \in \mathfrak{M}$ and m_i is a child of m in $\mathcal{T}'$.

For the second condition, let $(s', m) \in \mathfrak{M}$ where m is a modal node, meaning that

$$\text{(mod)}\ \frac{m = \langle \Gamma \cup \{N^1_{r_1}\varphi_1, \ldots, N^n_{r_n}\varphi_n\} \cup \{\neg O^1_{r'_1}\varphi'_1, \ldots, \neg O^{n'}_{r'_{n'}}\varphi'_{n'}\}, \mathcal{I}^L, \mathcal{I}^M\rangle}{m_1 = \langle \{\psi_1\}, \mathcal{I}^L_1, \mathcal{I}^M_1\rangle \quad \cdots \quad m_k = \langle \{\psi_k\}, \mathcal{I}^L_k, \mathcal{I}^M_k\rangle}$$

For every ψ_i we must have $N^j_{r_j}\varphi_j = N^j_{r_j}\psi_i$ for some j, so $\mathcal{M}, s' \models N^j_{r_j}\psi_i$, which implies that there exists $t' \in [\![\psi_i]\!]$ such that $s' \xrightarrow{r} t'$ for some r. Hence we get $(t', m_i) \in \mathfrak{M}$. Since this holds for any i, we get that every m_i is included in $\mathcal{T}'$.

For the third condition, let $m = \langle \Gamma, \mathcal{I}^L, \mathcal{I}^M\rangle$ be a terminal node in $\mathcal{T}'$. We check the conditions of Definition A.5.6. There must exist a state s' such that $(s', m) \in \mathfrak{M}$, which means that $\mathcal{M}, s' \models \Gamma$. Hence s' satisfies all the literals in Γ, which can only happen if the first condition is satisfied. The second and third condition are satisfied because of the way the intervals of the children are constructed in the (mod) rule.

($\Longleftarrow$) This follows from Lemma A.5.8. ■

Theorem A.5.11. *The satisfiability problem for our logic is decidable.*

Proof. By Lemma A.5.10, to decide whether a formula φ is satisfiable, it is enough to check whether there exists a successful tableau for φ. Furthermore, by Lemma A.5.9 it is enough to only check a single tableau for φ: If the tableau is successful, then all tableaux for φ are successful, and if it is not successful, then no tableau for φ is successful.

One can construct such a tableau for φ by applying the tableau rules of Table A.5.1 to the tuple $\langle \{\varphi\}, \emptyset, (0, 0)\rangle$ until no more rules can be applied. We will now argue that there is an effective procedure for constructing such a tableau by induction on the modal depth of φ.

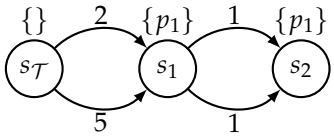


Figure A.5.1: The model $\mathcal{M}(\mathcal{T})$ for the successful tableau $\mathcal{T}$ in Example A.5.12.

$md(\varphi) = 0$: In this case, the (mod) rule is never used when constructing the tableau. Hence the procedure proceeds by syntactically checking which rules can be used at a given moment, and choosing a valid rule to apply.

$md(\varphi) > 0$: In this case we proceed as for the case where $md(\varphi) = 0$, except that now the (mod) rule may also be applied, in which case we need to be able to compute the ψ_i, Δ_i and (a_i, b_i). The difficulty lies in computing the set $\{\psi_1, \ldots, \psi_k\} = \min(\mathcal{L}(\{\varphi_1, \ldots, \varphi_n\}))$ and the sets

$$\mathbb{L}_i^- = \{r \mid L_r\varphi'_j = O^j_{r_j}\varphi'_j \text{ for some } j \text{ and } \models \psi_i \to \varphi'_j\}$$

$$\mathbb{M}_i^- = \{r \mid M_r\varphi'_j = O^j_{r_j}\varphi'_j \text{ for some } j \text{ and } \models \psi_i \to \varphi'_j\}.$$

However, note that all φ_i and φ'_i and have modal depth less than $md(\varphi)$. Therefore, by induction hypothesis, we have an effective procedure to decide whether $\models \varphi_i \to \varphi_j$ and $\models \varphi_i \leftrightarrow \varphi_j$, which is exactly what we need to compute the aforementioned sets. Given this we can compute the values needed for the intervals $\mathcal{I}_i^L$ and $\mathcal{I}_i^M$. ■

Example A.5.12. Consider the formula $\varphi = \neg(\neg(L_2p_1 \wedge M_5L_1p_1) \wedge \neg M_2p_2))$. Using the tableau rules, we get the following tableau $\mathcal{T}$ for φ.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle\{\neg(\neg(L_2p_1 \wedge M_5L_1p_1) \wedge M_2p_2)\}, [0,0], [0,0]\rangle}{\langle\{\neg\neg(L_2p_1 \wedge M_5L_1p_1)\}, [0,0], [0,0]\rangle}\,(\neg\wedge)}{\langle\{L_2p_1 \wedge M_5L_1p_1\}, [0,0], [0,0]\rangle}\,(\neg\neg)}{\langle\{L_2p_1, M_5L_1p_1\}, [0,0], [0,0]\rangle}\,(\wedge)}{\langle\{p_1, L_1p_1\}, [2,\infty), [5,\infty)\rangle}\,(\text{mod})}{\langle\{p_1\}, [1,\infty), [0,\infty)\rangle}\,(\text{mod}) \qquad \dfrac{\dfrac{\langle\{\neg\neg M_2p_2\}, [0,0], [0,0]\rangle}{\langle\{M_2p_2\}, [0,0], [0,0]\rangle}\,(\neg\neg)}{\langle\{p_2\}, [0,\infty), [0,2]\rangle}\,(\text{mod})$$

In this case the tableau is successful, since all terminal nodes are consistent. In fact, there are three distinct subtrees witnessing this fact: one that chooses the left branch, one that chooses the right branch, and one that chooses both branches. In Figure A.5.1 we show the resulting model $\mathcal{M}(\mathcal{T})$ for the witness that chooses the left branch. ♦

Example A.5.13. Consider the formula $\varphi = p_1 \wedge L_4 p_1 \wedge \neg L_3 p_1 \wedge L_2 p_2$. Using the tableau rules, we get the following tableau $\mathcal{T}$ for φ.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle\{p_1 \wedge L_4 p_1 \wedge \neg L_3 p_1 \wedge L_2 p_2\}, [0,0], [0,0]\rangle}{\langle\{p_1, L_4 p_1 \wedge \neg L_3 p_1 \wedge L_2 p_2\}, [0,0], [0,0]\rangle}\,(\wedge)}{\langle\{p_1, L_4 p_1, \neg L_3 p_1 \wedge L_2 p_2\}, [0,0], [0,0]\rangle}\,(\wedge)}{\langle\{p_1, L_4 p_1, \neg L_3 p_1, L_2 p_2\}, [0,0], [0,0]\rangle}\,(\wedge)}{\langle\{p_1\}, [4,3), [0,\infty]\rangle \quad \langle\{p_2\}, [2,\infty), [0,\infty)\rangle}\,(\text{mod})$$

In this case the interval $[4,3)$ is not consistent, and hence the tableau is not successful, so we can conclude that φ is not satisfiable. ♦

A.6 Concluding Remarks

Our contributions in this paper have been to define a new bisimulation relation for weighted transition systems (WTSs), which relates those states that have similar behavior with respect to their minimum and maximum weights on transitions, as well as an accompanying modal logic to reason about the upper and lower bounds of weights on transitions. We have shown that this logic characterises exactly those states that are bisimilar for image-finite systems. Furthermore, we have provided a complete axiomatisation of our logic, and we have shown that it enjoys the finite model property. Based on this finite model property, we have developed an algorithm which decides the satisfiability of a formula in our logic and constructs a finite model for the formula if it is satisfiable.

This work could be extended in different ways. Since our logic is non-compact, strong completeness does not follow directly from weak completeness, and hence it would be interesting to explore a strong-complete axiomatisation of the proposed logic. Such an axiomatisation would need additional, infinitary axioms. Examples of such axioms would be

$$\{L_q \varphi \mid q < r\} \vdash L_r \varphi \quad \text{and} \quad \{M_q \varphi \mid q < r\} \vdash M_r \varphi,$$

which are easily proven sound and describe the Archimedean property discussed in Theorem A.4.11.

Although we have shown that our logic is expressive enough to capture bisimulation, it would also be of interest to extend our logic with a kind of fixed-point operator or standard temporal logic operators such as until in order to increase its expressivity, and hence its practical use. We envisage two ways in which such a logic could be given semantics: either by accumulating weights or by taking the maximum or minimum of weights. In the accumulating case in particular, one could also allow negative weights to model that the system gains resources.

A.7 References

[1] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking in dense real-time," *Inf. Comput.*, vol. 104, no. 1, pp. 2–34, 1993. [Online]. Available: https://doi.org/10.1006/inco.1993.1024

[2] P. Babari, M. Droste, and V. Perevoshchikov, "Weighted register automata and weighted logic on data words," in *Theoretical Aspects of Computing - ICTAC 2016 - 13th International Colloquium, Taipei, Taiwan, ROC, October 24-31, 2016, Proceedings*, ser. Lecture Notes in Computer Science, A. Sampaio and F. Wang, Eds., vol. 9965, 2016, pp. 370–384. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46750-4_21

[3] P. Blackburn, J. F. A. K. van Benthem, and F. Wolter, *Handbook of Modal Logic*, ser. Studies in Logic and Practical Reasoning. Elsevier Science, 2006.

[4] B. Bollig and P. Gastin, "Weighted versus probabilistic logics," in *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, ser. Lecture Notes in Computer Science, V. Diekert and D. Nowotka, Eds., vol. 5583. Springer, 2009, pp. 18–38. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02737-6_2

[5] L. Cardelli, K. G. Larsen, and R. Mardare, "Continuous Markovian logic - from complete axiomatization to the metric space of formulas," in *Computer Science Logic, 25th International Workshop / 20th Annual Conference of the EACSL, CSL 2011, September 12-15, 2011, Bergen, Norway, Proceedings*, ser. LIPIcs, M. Bezem, Ed., vol. 12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011, pp. 144–158. [Online]. Available: http://dx.doi.org/10.4230/LIPIcs.CSL.2011.144

[6] ——, "Modular Markovian logic," in *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, ser. Lecture Notes in Computer Science, L. Aceto, M. Henzinger, and J. Sgall, Eds., vol. 6756. Springer, 2011, pp. 380–391. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22012-8_30

[7] S. Chakraborty and J. Katoen, "On the satisfiability of some simple probabilistic logics," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, M. Grohe, E. Koskinen, and N. Shankar, Eds. ACM, 2016, pp. 56–65. [Online]. Available: http://doi.acm.org/10.1145/2933575.2934526

[8] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Trans. Program. Lang. Syst.*, vol. 8, no. 2, pp. 244–263, 1986. [Online]. Available: http://doi.acm.org/10.1145/5397.5399

[9] M. Droste and P. Gastin, "Weighted automata and weighted logics," in *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580. Springer, 2005, pp. 513–525. [Online]. Available: http://dx.doi.org/10.1007/11523468_42

[10] M. Droste and G. Rahonis, "Weighted automata and weighted logics on infinite words," in *Developments in Language Theory, 10th International Conference, DLT 2006, Santa Barbara, CA, USA, June 26-29, 2006, Proceedings*, ser. Lecture Notes in Computer Science, O. H. Ibarra and Z. Dang, Eds., vol. 4036. Springer, 2006, pp. 49–58. [Online]. Available: http://dx.doi.org/10.1007/11779148_6

[11] M. Droste and H. Vogler, "Weighted tree automata and weighted logics," *Theor. Comput. Sci.*, vol. 366, no. 3, pp. 228–247, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2006.08.025

[12] Z. Ésik, "Axiomatizing weighted synchronization trees and weighted bisimilarity," *Theor. Comput. Sci.*, vol. 534, pp. 2–23, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2014.02.033

[13] R. Fagin and J. Y. Halpern, "Reasoning about knowledge and probability," *J. ACM*, vol. 41, no. 2, pp. 340–367, 1994. [Online]. Available: http://doi.acm.org/10.1145/174652.174658

[14] I. Fichtner, "Weighted picture automata and weighted logics," *Theory Comput. Syst.*, vol. 48, no. 1, pp. 48–78, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00224-009-9225-3

[15] S. Givant and P. Halmos, *Introduction to Boolean Algebras*, ser. Undergraduate Texts in Mathematics. Springer, 2009.

[16] M. Hansen, K. G. Larsen, R. Mardare, M. R. Pedersen, and B. Xue, "A complete approximation theory for weighted transition systems," in *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016, Beijing, China, November 9-11, 2016, Proceedings*, ser. Lecture Notes in Computer Science, M. Fränzle, D. Kapur, and N. Zhan, Eds., vol. 9984, 2016, pp. 213–228. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-47677-3_14

[17] A. Heifetz and P. Mongin, "Probability logic for type spaces," *Games and Economic Behavior*, vol. 35, no. 1-2, pp. 31–53, 2001. [Online]. Available: http://dx.doi.org/10.1006/game.1999.0788

[18] S. Jaziri, K. G. Larsen, R. Mardare, and B. Xue, "Adequacy and complete axiomatization for timed modal logic," *Electr. Notes Theor. Comput. Sci.*, vol. 308, pp. 183–210, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2014.10.011

[19] B. Jonsson and K. G. Larsen, "Specification and refinement of probabilistic processes," in *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*. IEEE Computer Society, 1991, pp. 266–277. [Online]. Available: https://doi.org/10.1109/LICS.1991.151651

[20] L. Juhl, K. G. Larsen, and J. Srba, "Modal transition systems with weight intervals," *J. Log. Algebr. Program.*, vol. 81, no. 4, pp. 408–421, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jlap.2012.03.008

[21] D. Kozen, K. G. Larsen, R. Mardare, and P. Panangaden, "Stone duality for Markov processes," in *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 2013, pp. 321–330. [Online]. Available: http://dx.doi.org/10.1109/LICS.2013.38

[22] D. Kozen, R. Mardare, and P. Panangaden, "Strong completeness for Markovian logics," in *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and J. Sgall, Eds., vol. 8087. Springer, 2013, pp. 655–666. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40313-2_58

[23] K. G. Larsen and R. Mardare, "Complete proof systems for weighted modal logic," *Theor. Comput. Sci.*, vol. 546, pp. 164–175, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2014.03.007

[24] K. G. Larsen, R. Mardare, and B. Xue, "Alternation-free weighted mu-calculus: Decidability and completeness," *Electr. Notes Theor. Comput. Sci.*, vol. 319, pp. 289–313, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2015.12.018

[25] ——, "Concurrent weighted logic," *J. Log. Algebr. Meth. Program.*, vol. 84, no. 6, pp. 884–897, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jlamp.2015.07.002

[26] ——, "On decidability of recursive weighted logics," *Soft Comput.*, vol. 22, no. 4, pp. 1085–1102, 2018. [Online]. Available: https://doi.org/10.1007/s00500-016-2193-z

[27] ——, "Decidability and expressiveness of recursive weighted logic," in *Perspectives of System Informatics - 9th International Ershov Informatics Conference, PSI 2014, St. Petersburg, Russia, June 24-27, 2014. Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Voronkov and I. Virbitskaite, Eds., vol. 8974. Springer, 2014, pp. 216–231. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-46823-4_18

[28] ——, "A decidable recursive logic for weighted transition systems," in *Theoretical Aspects of Computing - ICTAC 2014 - 11th International Colloquium, Bucharest, Romania, September 17-19, 2014. Proceedings*, ser. Lecture Notes in Computer Science, G. Ciobanu and D. Méry, Eds., vol. 8687. Springer, 2014, pp. 460–476. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10882-7_27

[29] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Inf. Comput.*, vol. 94, no. 1, pp. 1–28, 1991. [Online]. Available: https://doi.org/10.1016/0890-5401(91)90030-6

[30] R. Mardare, L. Cardelli, and K. G. Larsen, "Continuous Markovian logics - axiomatization and quantified metatheory," *Logical Methods in Computer Science*, vol. 8, no. 4, 2012. [Online]. Available: https://doi.org/10.2168/LMCS-8(4:19)2012

[31] I. Meinecke, "Weighted logics for traces," in *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, ser. Lecture Notes in Computer Science, D. Grigoriev, J. Harrison, and E. A. Hirsch, Eds., vol. 3967. Springer, 2006, pp. 235–246. [Online]. Available: http://dx.doi.org/10.1007/11753728_25

[32] C. Zhou, "A complete deductive system for probability logic," *J. Log. Comput.*, vol. 19, no. 6, pp. 1427–1454, 2009. [Online]. Available: http://dx.doi.org/10.1093/logcom/exp031

Paper B

Timed Comparisons of Semi-Markov Processes

Mathias R. Pedersen, Nathanaël Fijalkow, Giorgio Bacci, Kim G. Larsen, and Radu Mardare

The paper has been published in the
Proceedings of the 12th International Conference on Language and Automata Theory and Applications, pp. 271–283, 2018.

The layout has been revised and the content extended.

Abstract

Semi-Markov processes are Markovian processes in which the firing time of the transitions is modelled by probabilistic distributions over positive reals interpreted as the probability of firing a transition at a certain moment in time.

In this paper we consider the trace-based semantics of semi-Markov processes, and investigate the question of how to compare two semi-Markov processes with respect to their time-dependent behaviour. To this end, we introduce the relation of being "faster than" between processes and study its algorithmic complexity. Through a connection to probabilistic automata we obtain hardness results showing in particular that this relation is undecidable. However, we present an additive approximation algorithm for a time-bounded variant of the faster-than problem over semi-Markov processes with slow residence-time functions, and a **coNP** *algorithm for the exact faster-than problem over unambiguous semi-Markov processes. Finally, we give a logical characterisation of the faster-than relation and show that satisfiability and model checking are decidable for this logic.*

B.1 Introduction

Semi-Markov processes are Markovian stochastic systems that model the firing time of transitions as probabilistic distribution over positive reals; thus, one can encode the probability of firing a certain transition within a certain time interval. For example, continuous-time Markov processes are particular case of semi-Markov processes where the timing distributions are always exponential.

Semi-Markov processes have been used extensively to model real-time systems such as power plants [16] and power supply units [17]. For such real-time systems, non-functional requirements are becoming increasingly important. Many of these requirements, such as response time and throughput, depend heavily on the timing behaviour of the system in question. It is therefore natural to understand and be able to compare the timing behaviour of different systems.

Moller and Tofts [13] proposed the notion of a *faster-than* relation for systems with discrete-time in the context of process algebras. Their goal was to be able to compare processes that are functionally behaviourally equivalent, except that one process may execute actions faster than the other. This line of study was continued by Lüttgen and Vogler [12], who moreover considered upper bounds on time, in order to allow for reasoning about worst-case timing behaviours. For timed automata, Guha et al. [10] introduced a bisimulation-like faster-than relation and studied its compositional properties. For continuous-time probabilistic systems, Baier et al. [3] considered a simulation relation where the timing distribution on each state is required

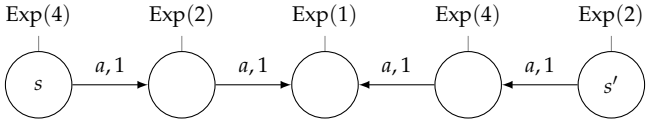


Figure B.1.1: A semi-Markov process where s is faster than s'. The states of the process are annotated with their timing distributions and each action-labelled transition is decorated with its probability to be executed.

to stochastically dominate the other. They introduced both a weak and a strong version of their simulation relation, and gave a logical characterisation of these in terms of the logic CSL.

In the literature, less attention has been drawn to trace-based notions of faster-than relations although trace equivalence and inclusion are important concepts when considering linear-time properties such as liveness or safety [2]. In this paper we propose a simple and intuitive notion of trace inclusion for semi-Markov processes, which we call *faster-than* relation, that compares the relative speed of processes with respect to the execution of arbitrary sequences of actions.

Differently from trace inclusion, our relation does not make a step-wise comparison of the timing delays for each individual action in a sequence, but over the overall execution time of the sequence. As an example, consider the semi-Markov process in Fig. B.1.1. The states s and s', although performing the same sequences of actions, are not related by trace inclusion because the first two actions in any sequence are individually executed at opposite order of speeds (here governed by exponential-time distributions). Instead, according to our relation, s *is* faster-than s' (but not vice versa) because it executes single-action sequences at a faster rate than s', and action sequences of length greater than one at the same speed – this is due to the fact that the execution time of each action is governed by random variables that are independent of each other and the sum of independent random variables is commutative.

In this paper we investigate the algorithmic complexity of various problems regarding the faster-than relation, emphasising their connection with classical algorithmic problems over Rabin's probabilistic automata. In particular, we prove that the faster-than problem over generic semi-Markov processes is undecidable and that it is Positivity-hard when restricted to processes with only one action label. The reduction from the Positivity problem is important because it relates the faster-than problem to the Skolem problem, an important problem in number theory, whose decidability status has been an open problem for at least 80 years [1, 14].

We show that undecidability for the faster-than problem can not be tackled even by approximation techniques: via the same connection with proba-

bilistic automata we are able to prove that the faster-than problem can not be approximated up to a multiplicative constant. However, as a positive result, we show that a time-bounded variant of the faster-than problem, which compares processes up to a given finite time bound, although still undecidable, admits approximated solutions up to an *additive* constant over semi-Markov processes with slow residence-time distributions. These include the important cases of uniform and exponential distributions. As a second positive result, we present a **coNP** algorithm for solving the faster-than problem exactly over unambiguous semi-Markov processes, where a process is unambiguous if every transition to a next state is unambiguously determined by the label that it outputs.

Finally, we give a logical characterisation of the faster-than relation in terms of a very simple logic. Every formula in this logic is satisfiable in some finite model, and hence the satisfiability problem for the logic is trivially decidable. Furthermore, we show that the model checking problem for the logic is also decidable for many common residence-time distributions.

B.2 Definitions

For a finite set S we let $\mathcal{D}(S)$ denote the set of (sub)distributions over S, i.e. functions $\delta : S \to [0,1]$ such that $\sum_{s \in S} \delta(s) \leq 1$. The subset of total distributions is $\mathcal{D}_{=1}(S)$.

We let $\mathbb{N}$ denote the natural numbers and $\mathbb{R}_{\geq 0}$ denote the non-negative real numbers. We equip $\mathbb{R}_{\geq 0}$ with the Borel σ-algebra $\mathcal{B}$, so that $(\mathbb{R}_{\geq 0}, \mathcal{B})$ is a measurable space. Let $\mathcal{D}(\mathbb{R}_{\geq 0})$ denote the set of (sub)distributions over $(\mathbb{R}_{\geq 0}, \mathcal{B})$, i.e. measures $\mu : \mathcal{B} \to [0,1]$ such that $\mu(\mathbb{R}_{\geq 0}) \leq 1$. Throughout the paper we will write $\mu(t)$ for $\mu([0,t])$.

To avoid confusion we will refer to μ in $\mathcal{D}(\mathbb{R}_{\geq 0})$ as timing distributions, and to δ in $\mathcal{D}(S)$ as distributions.

Definition B.2.1 (Semi-Markov process)**.** A *semi-Markov process*, usually written $\mathcal{M}$, is given by:

- S is a (finite) set of *states*,
- $\mathtt{Out}$ is a (finite) set of *output labels*,
- $\Delta : S \to \mathcal{D}(S \times \mathtt{Out})$ is a *transition function*,
- $\rho : S \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is a *residence-time function*. ▲

The operational behaviour of a semi-Markov process can be described as follows. In a given state $s \in S$, the process fires a transition within time t with probability $\rho(s)(t)$, leading to the state $s' \in S$ while outputting the label $a \in \mathtt{Out}$ with probability $\Delta(s)(s', a)$.

We aim at defining $\mathbb{P}_{\mathcal{M}}(s, w, t)$, the probability that from the state s, the output of the semi-Markov process $\mathcal{M}$ within time t *starts with* the word w. It is important to note here that time is accumulated: we sum together the time spent in all states along the way, and ask that this total time is less than the specified bound t. A full and formal definition of the probability can be done through the usual cylinder construction. However, we will spare the reader this well-known construction and give seemingly ad-hoc definitions in this conference version.

In order to account for the accumulated time in the probability, we need the notion of convolution. The convolution of two timing distributions μ and ν is $\mu * \nu$ defined by

$$(\mu * \nu)(E) = \int_0^\infty \nu(E - x)\mu(\mathrm{d}x)$$

for any Borel set E. Convolution is both associative and commutative. Let X and Y be two independent random variables with timing distributions μ and ν, i.e. $\mathbb{P}(X \in E) = \mu(E)$ and $\mathbb{P}(Y \in E) = \nu(E)$, then

$$\mathbb{P}(X + Y \in E) = (\mu * \nu)(E).$$

Definition B.2.2 (Probability)**.** Consider a semi-Markov process $\mathcal{M}$. We define the timing distribution $\mathbb{P}_{\mathcal{M}}(s, w)$ inductively: $\mathbb{P}_{\mathcal{M}}(s, \varepsilon) = \mathbb{1}$ for the empty word ε, where $\mathbb{1}$ is the function such that $\mathbb{1}(t) = 1$ for all t in $\mathbb{R}_{\geq 0}$, and for a word w in $\mathtt{Out}^*$, a letter a in $\mathtt{Out}$ and a state s,

$$\mathbb{P}_{\mathcal{M}}(s, aw) = \sum_{s' \in S} \Delta(s)(s', a) \cdot \left(\rho(s) * \mathbb{P}_{\mathcal{M}}(s', w)\right).$$

We will then write $\mathbb{P}_{\mathcal{M}}(s, w, t)$ to mean $\mathbb{P}_{\mathcal{M}}(s, w)(t)$. ▲

B.2.1 Timed Comparisons

We introduce the following relation which will be the focus of our paper.

Definition B.2.3 (Faster-than relation)**.** Consider a semi-Markov process $\mathcal{M}$ and two states s and s'. We say that s is *faster than* s', denoted $s \preceq s'$, if for all w, for all t,

$$\mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t).$$ ▲

The algorithmic problem we consider in this paper is the *faster-than problem*: given a semi-Markov process and two states s and s', determine whether $s \preceq s'$.

B.2.2 Algorithmic Considerations

The definition we use for semi-Markov processes is very general, because we allow for any residence-time function. The aim of the paper is to give generic algorithmic results which apply to *effective* classes of timing distributions, a notion we define now. Recall that a residence-time function associates with each state a timing distribution. We first give some examples of classical timing distributions.

- The prime example is exponential distributions, defined by the timing distribution $\mu(t) = 1 - e^{-\lambda t}$ for some parameter $\lambda > 0$ usually called the rate.
- Another interesting example is that of piecewise polynomial distributions. Consider finitely many polynomials $P_1, \ldots, P_n$ and a finite set of pairwise disjoint intervals $I_1 \cup I_2 \cup \cdots \cup I_n$ covering $[0, \infty)$ such that for every k, P_k is non-negative over I_k and $\sum_k \int_{I_k} P_k = 1$. This induces the timing distribution
$$\mu(t) = \sum_k \int_{I_k \cap [0,t]} P_k(t).$$
- A special case of the previous example is given by piecewise affine distributions, where the polynomials are affine functions.
- Another important special case of piecewise polynomial distributions are the uniform distributions with parameters $0 \leq a < b$ defining the timing distribution
$$\mu(t) = \begin{cases} 1 & \text{if } t < a, \\ \frac{t-a}{b-a} & \text{if } t \in [a, b) \\ 0 & \text{if } x \geq b. \end{cases}$$
- The simplest example is given by Dirac distributions defined for the parameter a by $\mu(E) = 1$ if a is in E, and 0 otherwise.

The following definition captures these examples, and more. For a class $\mathcal{C}$ of timing distributions, we let $\mathrm{Convex}(\mathcal{C})$ be the smallest class of timing distributions containing $\mathcal{C}$ and closed under convex combinations, and similarly $\mathrm{Conv}(\mathcal{C})$ adding closure under convolutions.

Lemma B.2.4. *Let $\mathcal{C}$ be a class of timing distributions. Consider a semi-Markov process $\mathcal{M}$ whose residence-time function uses timing distributions from $\mathcal{C}$, a state s and a word w, then $\mathbb{P}_{\mathcal{M}}(s, w) \in \mathrm{Conv}(\mathcal{C})$.*

Lemma B.2.4 is established by a straightforward induction on the word w using the definition of $\mathbb{P}_{\mathcal{M}}(s, w)$.

In the rest of the paper we will consider only distributions that are suitable for algorithmic manipulation. Clearly, we must be able to give them as input to a computational device, so we assume they can be described by finitely many rational parameters. Moreover, we require that testing inequalities between them is decidable, since this is essential for determining the faster-than relation. The next definition formalises this intuition.

Definition B.2.5 (Effective timing distributions)**.** A class $\mathcal{C}$ of timing distributions is *effective* if, for any $\varepsilon \geq 0$, $b \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and $\mu_1, \mu_2 \in \mathrm{Conv}(\mathcal{C})$, it is decidable whether $\mu_1(t) \geq \mu_2(t) - \varepsilon$, for all $t \leq b$. ▲

Many common classes of timing distributions are effective, and can be decided using the existential theory of the reals. To show this, we make use of the following lemma.

Lemma B.2.6. *Let $F_1, F_2 : \mathbb{R}_{\geq 0} \to [0,1]$ be given timing distributions. If there exists a surjective function $T : [0,1] \to \mathbb{R}_{\geq 0}$ such that the functions $F_1 \circ T$ and $F_2 \circ T$ are semialgebraic, then it is decidable whether $F_1(t) \geq F_2(t)$ for all $t \in \mathbb{R}_{\geq 0}$.*

Proof. Since $F_1 \circ T$ and $F_2 \circ T$ are semialgebraic, the formula φ defined by

$$\varphi = \forall t.((0 \leq t \leq 1) \implies F_1(T(t)) \geq F_2(T(t)))$$

is expressible in the existential theory of the reals (or rather, its negation is), so we can decide whether φ is true by exploiting the decidability of the existential theory of the reals [18]. We now claim that $F_1(t) \geq F_2(t)$ for all t is true if and only if φ is true.

Assume that $F_1(t) \geq F_2(t)$ for all $t \in \mathbb{R}_{\geq 0}$ is true. Pick an arbitrary $t' \in [0,1]$. Then $T(t') \in \mathbb{R}_{\geq 0}$, so we know that $F_1(T(t')) \geq F_2(T(t'))$, and hence φ is true.

Next assume that φ is true and pick an arbitrary $t' \in \mathbb{R}_{\geq 0}$. Because T is surjective, there must be some $y \in [0,1]$ such that $t' = T(y)$. Hence we know that

$$F_1(t') = F_1(T(y)) \geq F_2(T(y)) = F_2(t').$$ ■

In Lemma B.2.6, T is a *transformation* or *variable change* which turns the given functions into piecewise polynomial functions. The requirement that the transformation be surjective ensures that deciding the inequality between the transformed functions is equivalent to deciding it between the original functions.

Proposition B.2.7. *The following classes of timing distributions are effective:*

- *exponential distributions with rational rates,*
- *piecewise polynomial distributions,*

- *piecewise affine distributions,*
- *uniform distributions,*
- *Dirac distributions.*

Proof. Let $\mathcal{C}$ be a class of timing distributions. We want to decide whether

$$(\mu_1 * \cdots * \mu_n)(t) \geq (\nu_1 * \cdots * \nu_n)(t) \quad \text{for all } t \in \mathbb{R}_{\geq 0}$$

whenever $\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n \in \mathcal{C}$. If we let $F_1(t) = (\mu_1 * \cdots * \mu_n)(t)$ and $F_2(t) = (\nu_1 * \cdots * \nu_n)(t)$, then Lemma B.2.6 tells us that we can decide whether $F_1(t) \geq F_2(t)$ for all $t \in \mathbb{R}_{\geq 0}$ by finding an appropriate surjective function $T : [0,1] \to \mathbb{R}_{\geq 0}$.

When $\mathcal{C}$ is either the class of Dirac distributions, the class of piecewise affine distributions, or the class of piecewise polynomial distributions, this is trivial, since these are all already semialgebraic. Hence we can simply take T to be the identity function. Although it is perhaps less obvious, the same is also true when $\mathcal{C}$ is the class of uniform distributions [11].

This leaves the case when $\mathcal{C}$ is the class of exponential distributions with rational rates. Assume that each μ_i has rate λ_i and each ν_i has rate λ_i'. It was shown in [4] that F_1 has the following closed form. Assume that there are m distinct rates among $\lambda_1, \ldots, \lambda_n$ and reorder $\lambda_1, \ldots, \lambda_n$ such that $\lambda_1, \ldots, \lambda_{r_1}$ are identical, $\lambda_{r_1+1}, \ldots, \lambda_{r_1+r_2}$ are identical, and so forth. This reordering does not change the values of F_1 because convolution is both associative and commutative. Now let $\alpha_1 = \lambda_{r_1}, \alpha_2 = \lambda_{r_1+r_2}, \ldots, \alpha_m = \lambda_{r_1+\cdots+r_m}$. The closed form of F_1 is then given by

$$F_1(t) = 1 - \sum_{k=1}^{m} \sum_{l=1}^{r_k} C_1(k,l) \cdot e^{-\lambda_k \cdot t},$$

where $C_1(k,l)$ is an expression that depends on k and l, but not on t. Note also that $C_1(k,l)$ is expressible in the existential theory of the reals. Likewise, F_2 will have a closed form

$$F_2(t) = 1 - \sum_{k=1}^{m'} \sum_{l=1}^{r_k'} C_2(k,l) \cdot e^{-\beta_k \cdot t}.$$

Now let $T : [0,1] \to \mathbb{R}_{\geq 0}$ be given by

$$T(x) = \begin{cases} 0 & \text{if } x = 0 \\ -\frac{1}{\gcd(\lambda_1, \ldots, \lambda_n, \lambda_1', \ldots, \lambda_n')} \cdot \ln(x) & \text{otherwise.} \end{cases}$$

For convenience, we let $\theta = \gcd(\lambda_1, \ldots, \lambda_n, \lambda_1', \ldots, \lambda_n')$. Then T is surjective, for if $y \in \mathbb{R}_{\geq 0}$, then $x = e^{-\theta \cdot y} \in [0,1]$ and $T(x) = y$. Furthermore,

$$F_1(T(x)) = 1 - \sum_{k=1}^{m} \sum_{l=1}^{r_k} C_1(k,l) \cdot x^{\frac{\alpha_k}{\theta}},$$

and because θ divides α_k for each k, it follows that $F_1 \circ T$ is a polynomial and hence semialgebraic. In a similar fashion we can show that $F_2 \circ T$ is also semialgebraic. ■

Proposition B.2.7 relies on decidability results for the existential theory of the reals [5, 18], implying that the most demanding operations above can be performed in polynomial space.

An effective class $\mathcal{C}$ of timing distributions induces the set of semi-Markov processes whose residence-time functions use timing distributions from $\mathcal{C}$. Furthermore, a given semi-Markov process has only finitely many states, and hence can only use finitely many timing distributions. For our decidability results we will therefore focus on finite classes of timing distributions. This paper gives algorithmic results for generic effective classes of timing distributions. In our complexity analyses, we will always assume that the operations on the timing distributions have a unit cost.

B.3 Hardness Results

We start the technical part of this article by hardness results inherited from Markov processes. A Markov process is a semi-Markov process without the residence-time function, and for a Markov process $\mathcal{M} = (S, \mathtt{Out}, \Delta)$, we define the probability

$$\mathbb{P}_{\mathcal{M}}(s, aw) = \sum_{s' \in S} \Delta(s)(s', a) \cdot \mathbb{P}_{\mathcal{M}}(s')(w)$$

and

$$\mathbb{P}_{\mathcal{M}}(s, \varepsilon) = 1$$

for the empty word. The faster-than relation for Markov processes is then $s \preceq s'$ if for all w we have $\mathbb{P}_{\mathcal{M}}(s, w) \geq \mathbb{P}_{\mathcal{M}}(s', w)$.

We show that the faster-than problem for Markov processes, and hence also for semi-Markov processes, is undecidable in general, can not be multiplicatively approximated, and relates to an open problem in number theory even in a restricted case. These limitations shape and motivate our positive results, which will be the topic of the remaining sections.

We first explain how hardness results for Markov processes directly imply hardness results for semi-Markov processes. The following lemma formalises the two ways semi-Markov processes subsume Markov processes.

Lemma B.3.1. *Consider a semi-Markov process $\mathcal{M} = (S, \mathtt{Out}, \Delta, \rho)$ and its induced Markov process $\mathcal{M}' = (S, \mathtt{Out}, \Delta)$.*

- *If ρ is constant, i.e. for all s, s' we have $\rho(s) = \rho(s')$, then for all w, for all t, we have $\mathbb{P}_{\mathcal{M}}(s, w, t) = \mathbb{P}_{\mathcal{M}'}(s, w) \cdot (\underbrace{\rho(s) * \cdots * \rho(s)}_{|w| \text{ times}})(t)$.*

- *If for all s, $\rho(s)$ is the Dirac distribution for 0, then for all w, for all t, we have $\mathbb{P}_{\mathcal{M}}(s, w, t) = \mathbb{P}_{\mathcal{M}'}(s, w)$.*

In particular in both cases, the following holds: for s, s' two states, we have $s \preceq s'$ in $\mathcal{M}$ if, and only if, $s \preceq s'$ in $\mathcal{M}'$.

We will use Lemma B.3.1 to draw corollaries about semi-Markov processes from hardness results of Markov processes.

The hardness results of this section will be based on a connection to probabilistic automata. A probabilistic automaton is given by

$$\mathcal{A} = (Q, A, q_0, \Delta : Q \times A \to \mathcal{D}_{=1}(Q), F),$$

where Q is the state space, A is the alphabet, q_0 is an initial state, Δ is the transition function, and F is a set of final or accepting states. Any probabilistic automaton $\mathcal{A}$ induces the probability $\mathbb{P}_{\mathcal{A}}(w)$ that a run over $w \in A^*$ is accepting, i.e. starts in q_0 and ends in F. The key property of probabilistic automata that we will exploit is the undecidability of the universality problem, which was proved in [15], see also [9]. The universality problem is as follows: given a probabilistic automaton $\mathcal{A}$, determine whether for all words w in A^+ we have $\mathbb{P}_{\mathcal{A}}(w) \geq \frac{1}{2}$.

We describe a construction which given a probabilistic automaton $\mathcal{A}$, constructs the *derived* Markov process $\mathcal{M}(\mathcal{A})$. The set of states of $\mathcal{M}(\mathcal{A})$ is $Q \times \{\ell, r\} \cup \{\top\}$, where $\top$ is a new state. Let $s = (q_0, \ell)$ and $s' = (q_0, r)$, where q_0 is the initial state of $\mathcal{A}$. The set of output labels is A, and the transition function Δ' is defined as follows:

$$\Delta'((p,\ell))((q,\ell),a) = \frac{1}{2|A|}\Delta(p,a)(q) \qquad \Delta'((p,\ell))(\top,a) = \frac{1}{2|A|} \text{ if } p \in F$$
$$\Delta'((p,r))((q,r),a) = \frac{1}{2|A|}\Delta(p,a)(q) \qquad \Delta'((p,r))(\top,a) = \frac{1}{4|A|}.$$

Lemma B.3.2.

$$\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s, wa) = \frac{1}{(2|A|)^{|w|+1}}\left(1 + \mathbb{P}_{\mathcal{A}}(w)\right)$$

and

$$\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s', wa) = \frac{1}{(2|A|)^{|w|+1}}\left(1 + \frac{1}{2}\right).$$

Proof. First observe that it can easily be proven by induction that

$$\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s, w) = \sum_{s_1 \in S} \cdots \sum_{s_n \in S} \Delta'(s)(s_1, w_1) \cdots \Delta'(s_{n-1})(s_n, w_n)$$

where $w = w_1 \dots w_n$ by simply unfolding the inductive definition of $\mathbb{P}$.

For the first equality, we therefore have

$$\begin{aligned}
&\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s, wa) \\
&= \sum_{s_1 \in S} \cdots \sum_{s_{n+1} \in S} \Delta'(s)(s_1, w_1) \cdots \Delta'(s_{n-1})(s_n, w_n) \cdot \Delta'(s_n)(s_{n+1}, a) \\
&= \sum_{s_1 \in Q \times \{\ell\}} \cdots \sum_{s_{n+1} \in Q \times \{\ell\}} \frac{1}{2|A|} \Delta(s, w_1)(s_1) \cdots \frac{1}{2|A|} \Delta(s_n, a)(s_{n+1}) \\
&\quad + \sum_{s_1 \in Q \times \{\ell\}} \cdots \sum_{s_n \in Q \times \{\ell\}} \frac{1}{2|A|} \Delta(s, w_1)(s_1) \cdots \Delta'(s_n)(\top, a) \\
&= \frac{1}{(2|A|)^{|w|+1}} + \frac{1}{(2|A|)^{|w|+1}} \cdot \mathbb{P}_{\mathcal{A}}(w) \\
&= \frac{1}{(2|A|)^{|w|+1}} \left(1 + \mathbb{P}_{\mathcal{A}}(w)\right).
\end{aligned}$$

For the second equality, we get

$$\begin{aligned}
&\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s', wa) \\
&= \sum_{s_1 \in S} \cdots \sum_{s_{n+1} \in S} \Delta'(s)(s_1, w_1) \cdots \Delta'(s_{n-1})(s_n, w_n) \cdot \Delta'(s_n)(s_{n+1}, a) \\
&= \sum_{s_1 \in Q \times \{r\}} \cdots \sum_{s_{n+1} \in Q \times \{r\}} \frac{1}{2|A|} \Delta(s, w_1)(s_1) \cdots \frac{1}{2|A|} \Delta(s_n, a)(s_{n+1}) \\
&\quad + \sum_{s_1 \in Q \times \{r\}} \cdots \sum_{s_n \in Q \times \{r\}} \frac{1}{2|A|} \Delta(s, w_1)(s_1) \cdots \frac{1}{4|A|} \\
&= \frac{1}{(2|A|)^{|w|+1}} + \frac{1}{(2|A|)^{|w|}} \cdot \frac{1}{4|A|} \\
&= \frac{1}{(2|A|)^{|w|+1}} \left(1 + \frac{1}{2}\right).
\end{aligned}$$

■

Theorem B.3.3. *The faster-than problem is undecidable for Markov processes.*

Proof. Given a probabilistic automaton $\mathcal{A}$, we construct the derived Markov process $\mathcal{M}(\mathcal{A})$. Thanks to the equalities in Lemma B.3.2, $\mathcal{A}$ is universal if, and only if, $s \preceq s'$. ■

We discuss three approaches to recover decidability.

A first approach is to look for *structural restrictions* on the underlying graph. However, the undecidability result above for probabilistic automata is quite robust in this aspect, as it already applies when the underlying graph is acyclic, meaning that the only loops are self-loops. In spite of this, we present in Section B.5 an algorithm to solve the faster-than problem for *unambiguous* semi-Markov processes.

A second approach is to restrict the *observations*. The undecidability result above holds already when there are two different output letters, hence a natural question is to look at what happens when we only have one output letter. Interestingly, specialising the construction above yields a reduction from the Positivity problem. This problem appears in various contexts, prominently in number theory, and its decidability status has been an open problem for at least 30 years [14]. Formally, the Positivity problem reads: given a linear recurrence sequence, are all terms of the sequence non-negative? It has been shown that the universality problem for probabilistic automata with one letter alphabet is equivalent to the Positivity problem [1]. Thus, using again the derived Markov process $\mathcal{M}(\mathcal{A})$ for a probabilistic automaton $\mathcal{A}$ with only one label, we obtain the following result.

Theorem B.3.4. *The faster-than problem is Positivity-hard over Markov processes with one output label.*

A third approach is *approximations*. However, we can exploit further the connection we made with probabilistic automata, obtaining an impossibility result for *multiplicative approximation*. We rely on the following celebrated theorem for probabilistic automata due to Condon and Lipton [6]. The following formulation of their theorem is described in detail in [7].

Theorem B.3.5 ([6])**.** *Let $0 < \alpha < \beta < 1$ be two constants. There is no algorithm which, given a probabilistic automaton $\mathcal{A}$,*

- *if for all w we have $\mathbb{P}_{\mathcal{A}}(w) \geq \beta$, returns YES,*
- *if there exists w such that $\mathbb{P}_{\mathcal{A}}(w) \leq \alpha$, returns NO.*

Theorem B.3.6. *Let $0 < \varepsilon < \frac{1}{3}$ be a constant. There is no algorithm which, given a Markov process $\mathcal{M}$ and two states s, s',*

- *if for all w we have $\mathbb{P}_{\mathcal{M}}(s, w) \geq \mathbb{P}_{\mathcal{M}}(s', w)$, returns YES,*
- *if there exists w such that $\mathbb{P}_{\mathcal{M}}(s, w) \leq \mathbb{P}_{\mathcal{M}}(s', w) \cdot (1 - \varepsilon)$, returns NO.*

Proof. Assume towards a contradiction that there exists an algorithm as described in the theorem. We then construct an algorithm satisfying the specifications of Theorem B.3.5.

Let $\alpha = \frac{1}{2} - \frac{3\varepsilon}{2}$ and $\beta = \frac{1}{2}$, and let $\mathcal{A}$ be a probabilistic automaton. We now run the algorithm on the derived Markov process $\mathcal{M}(\mathcal{A})$.

- If for all w we have $\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s, w) \geq \mathbb{P}_{\mathcal{M}(\mathcal{A})}(s', w)$, then the algorithm returns YES. Indeed, this is equivalent to $\mathbb{P}_{\mathcal{A}}(w) \geq \beta$.
- If there exists w such that $\mathbb{P}_{\mathcal{M}(\mathcal{A})}(s, w) \leq \mathbb{P}_{\mathcal{M}(\mathcal{A})}(s', w) \cdot (1 - \varepsilon)$, then the algorithm returns NO. Indeed, this is equivalent to $\mathbb{P}_{\mathcal{A}}(w) \leq \alpha$.

Hence we constructed an algorithm satisfying the specifications of Theorem B.3.5, a contradiction. ■

These hardness results for Markov processes together with Lemma B.3.1, gives us the following hardness results for semi-Markov processes.

Corollary B.3.7. *The following holds for semi-Markov processes for any class of timing distributions.*

- *The faster-than problem is undecidable.*
- *The faster-than problem with only one output label is Positivity-hard.*
- *The faster-than problem can not be multiplicatively approximated.*

B.4 Time-Bounded Additive Approximation

Instead of considering multiplicative approximation, we can also consider additive approximation, meaning that we want to decide whether for all w and t we have $\mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t) - \varepsilon$ for some constant $\varepsilon > 0$. In this section, we present an algorithm to solve the problem of approximating additively the faster-than relation with two assumptions:

- *time-bounded*: we only look at the behaviours up to a given bound b in $\mathbb{R}_{\geq 0}$,
- *slow residence-time functions*: each transition takes *some* time to fire.

As we will show, the combination of these two assumptions imply that the relevant words have bounded length. This is in contrast to the impossibility of approximating the faster-than relation multiplicatively that we showed in Sect. B.3. More precisely, we consider the *time-bounded* variant of the faster-than problem: given a time bound b in $\mathbb{R}_{\geq 0}$, a semi-Markov process, and two states s and s', determine whether for all $t \leq b$ and w it holds that $\mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t)$.

We first observe that this restriction of the faster-than problem does not make any of the problems in Sect. B.3 easier for semi-Markov processes. Indeed, if the residence-time functions are all Dirac distributions on 0, then all transitions are fired instantaneously, and the time-bounded restriction is immaterial. Thus we focus on distributions that do not fire instantaneously, as made precise by the following definition.

Definition B.4.1 (Slow distributions)**.** We say that a class $\mathcal{C}$ of timing distributions is *slow* if for all finite subset $\mathcal{C}_0$ of $\mathcal{C}$, there exists a computable function $\varepsilon \colon \mathbb{N} \times \mathbb{R}_{\geq 0} \to [0, 1]$ such that for all n, t, and $\mu_1, \ldots, \mu_n \in \mathrm{Convex}(\mathcal{C}_0)$ we have $(\mu_1 * \cdots * \mu_n)(t) \leq \varepsilon(n, t)$ and $\lim_{n \to \infty} \varepsilon(n, t) = 0$. ▲

Given a slow and effective class $\mathcal{C}$ of timing distributions, we can do additive approximation of the time-bounded faster-than problem in the following way. We introduce the following notation. Fix a semi-Markov process $\mathcal{M}$. Let $\mathcal{C}_{\mathcal{M}} = \text{Convex}(\{\rho(s) \mid s \in S\})$, and n in $\mathbb{N}$. We define the timing distribution $F_{\mathcal{M},n}$ by $F_{\mathcal{M},n}(t) = 1$ if $n = 0$ and otherwise

$$F_{\mathcal{M},n}(t) = \sup \{(\mu_1 * \cdots * \mu_n)(t) \mid \mu_1, \ldots, \mu_n \in \mathcal{C}_{\mathcal{M}}\}.$$

Lemma B.4.2. *For all s and all w, we have $\mathbb{P}_{\mathcal{M}}(s, w) \leq F_{\mathcal{M},|w|}$.*

Proof. We proceed by induction on the length of w. It is clear for $|w| = 0$.

$$\begin{aligned}
\mathbb{P}_{\mathcal{M}}(s, aw) &= \sum_{s' \in S} \Delta(s)(s', a) \cdot \rho(s) * \mathbb{P}_{\mathcal{M}}(s', w) \\
&\leq \underbrace{\sum_{s' \in S} \Delta(s)(s', a) \cdot \rho(s)}_{\in \mathcal{C}_{\mathcal{M}}} * F_{\mathcal{M},|w|} \\
&\leq F_{\mathcal{M},|w|+1}.
\end{aligned}$$

This concludes. ∎

Theorem B.4.3. *There exists an additive approximation algorithm for the time-bounded faster-than problem over semi-Markov processes for all slow and effective classes of timing distributions.*

In other words, for a constant $\varepsilon > 0$, there exists an algorithm which, given a semi-Markov process $\mathcal{M}$, two states s, s', and a bound b in $\mathbb{R}_{\geq 0}$, determines whether

$$\forall w, \forall t \leq b,\ \mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t) - \varepsilon.$$

Proof. Let $\mathcal{C}_{\mathcal{M}} = \text{Convex}(\{\rho(s) \mid s \in S\})$, since S is finite there exists a computable function $\varepsilon \colon \mathbb{N} \times \mathbb{R}_{\geq 0} \to [0, 1]$ such that for all n, t, and $\mu_1, \ldots, \mu_n \in \mathcal{C}_{\mathcal{M}}$ we have $(\mu_1 * \cdots * \mu_n)(t) \leq \varepsilon(n, t)$ and $\lim_{n \to \infty} \varepsilon(n, t) = 0$. Given $\varepsilon > 0$, there exists N such that $\varepsilon(N, b) < \varepsilon$. Let $n \geq N$. By assumption

$$(\mu_1 * \cdots * \mu_n)(b) \leq \varepsilon(n, b) \leq \varepsilon(N, b) < \varepsilon$$

for all $\mu_1, \ldots, \mu_n \in \mathcal{C}_{\mathcal{M}}$. Taking the supremum over $\mu_1, \ldots, \mu_n$, we then get $F_{\mathcal{M},n}(b) < \varepsilon$, and by Lemma B.4.2, this means that for all w of length at least N, we have $\mathbb{P}_{\mathcal{M}}(s', w, b) < \varepsilon$. Hence it holds trivially that for all $t \leq b$ and w of length at least N, we have $\mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t) - \varepsilon$.

Thus the algorithm checks whether for all words of length less than N, for all $t \leq b$, we have $\mathbb{P}_{\mathcal{M}}(s, w, t) \geq \mathbb{P}_{\mathcal{M}}(s', w, t) - \varepsilon$, which is decidable thanks to the effectiveness of $\mathcal{C}$. ∎

Next we show that there are interesting classes of timing distributions that are indeed slow. For this we introduce a class of timing distributions that are

not just slow, but furthermore are guaranteed to converge to zero rapidly. We say that a timing distribution μ is *very slow* if there exists a computable function $\varepsilon : \mathbb{R}_{\geq 0} \to [0,1]$ such that $\lim_{t\to 0} \frac{\varepsilon(t)}{t} = 0$ and for all t, we have $\mu(t) \leq \varepsilon(t)$. In order to show that very slow timing distributions are slow, we need the following lemma.

Lemma B.4.4. *Let $\mu_1, \ldots, \mu_n$ be timing distributions. Then*

$$(\mu_1 * \mu_2 * \cdots * \mu_n)(t) \leq \sum_{i=1}^{n} \mu_i\left(\frac{t}{n}\right).$$

Proof. We proceed by induction on n. The case of $n = 1$ is trivial. Recall that for any non-negative function f and measure μ we have

$$\int_E f(x)\mu(\mathrm{d}x) \leq \mu(E) \cdot (\sup_E f(x)). \tag{B.1}$$

Let $\mu = \mu_1 * \cdots * \mu_n$.

$$\begin{aligned}
&(\mu_1 * \cdots * \mu_{n+1})(t) \\
&= \int_0^t \mu(t-x)\mu_{n+1}(\mathrm{d}x) \\
&= \int_0^{\frac{nt}{n+1}} \mu(t-x)\mu_{n+1}(\mathrm{d}x) + \int_{\frac{nt}{n+1}}^t \mu(t-x)\mu_{n+1}(\mathrm{d}x) \\
&= \int_0^{\frac{nt}{n+1}} \mu(t-x)\mu_{n+1}(\mathrm{d}x) + \int_0^{\frac{t}{n+1}} \mu\left(\frac{t}{n+1} - u\right)\mu_{n+1}(\mathrm{d}u) \\
&\leq \mu\left(\frac{nt}{n+1}\right) + \mu_{n+1}\left(t - \frac{nt}{n+1}\right) \\
&\leq \sum_{i=1}^{n} \mu_i\left(\frac{n}{n+1}\frac{t}{n}\right) + \mu_{n+1}\left(\frac{t}{n+1}\right) = \sum_{i=1}^{n+1} \mu_i\left(\frac{t}{n+1}\right).
\end{aligned}$$

The third equality is the change of variable $u = x - \frac{nt}{n+1}$. The first inequality uses for each summand the inequality (B.1). The second inequality is by induction hypothesis. ∎

We can now prove the following theorem.

Theorem B.4.5. *The following classes of timing distributions are slow:*

- *very slow distributions,*
- *uniform distributions, and*
- *exponential distributions.*

Proof. Let $\mathcal{C}$ be a class of very slow timing distributions, and

$$\mathcal{C}_0 = \{\mu_1, \ldots, \mu_n\}$$

a finite subset of $\mathcal{C}$. Since every timing distribution in $\mathcal{C}$ is very slow, for every $i \in \{1, \ldots, n\}$ there exists a function ε_i such that $\mu_i(t) \leq \varepsilon_i(t)$ for all t. Let $\varepsilon(n,t) = n \cdot \max\left\{\varepsilon_i\left(\frac{t}{n}\right) \mid i \in \{1, \ldots, n\}\right\}$. Note that $\lim_{n\to\infty} \varepsilon(n,t) = 0$. Let $\nu_1, \ldots, \nu_n$ in $\mathrm{Convex}(\mathcal{C}_0)$, we have $(\nu_1 * \cdots * \nu_n)(t) \leq \sum_{i=1}^{n} \nu_i\left(\frac{t}{n}\right)$ thanks to Lemma B.4.4. This implies that $(\nu_1 * \cdots * \nu_n)(t) \leq \varepsilon(n,t)$, which concludes.

For exponential distributions, we proceed as follows. Let $\mathcal{C}_0$ be a finite class of exponential distributions. Let $\lambda > 0$ be the rate of the slowest exponential distributions appearing in $\mathcal{C}_0$, and let $\mu(t) = 1 - e^{-\lambda t}$. Then for any $\mu_1, \ldots, \mu_n$ in $\mathrm{Convex}(\mathcal{C}_0)$ we have

$$(\mu_1 * \cdots * \mu_n)(t) \leq (\underbrace{\mu * \cdots * \mu}_{n \text{ times}})(t).$$

The distribution $\mu * \cdots * \mu$ is called the Gamma (or more precisely, Erlang) distribution, and there is a computable closed form for it. In particular, if we let

$$\varepsilon(n,t) = (\underbrace{\mu * \cdots * \mu}_{n \text{ times}})(t),$$

we have $\lim_{n\to\infty} \varepsilon(n,b) = 0$, so exponential distributions are slow.

Uniform distributions can be handled using a similar way as for exponential distributions. Let $\mathcal{C}_0$ be a finite class of uniform distributions with parameters a_i and b_i for $i \in \{1, \ldots, n\}$. Let a be the smallest a_i and b the smallest b_i, and let μ be the uniform distribution with parameters a and b. Then it follows that

$$(\mu_1 * \cdots * \mu_n)(t) \leq (\underbrace{\mu * \cdots * \mu}_{n \text{ times}})(t) = \varepsilon(n,t).$$

Then $(\mu * \cdots * \mu)$ also has a nice closed form [11] and $\lim_{n\to\infty} \varepsilon(n,b) = 0$. ■

B.5 Unambiguous Semi-Markov Processes

In order to regain decidability of the faster-than relation, we can look at structurally simpler special cases of semi-Markov processes. Here we will focus on semi-Markov processes such that each output word induces at most one trace of states. More precisely, we will say that a semi-Markov process is *unambiguous* if for every s in S and a in $\mathtt{Out}$, there exists at most one s' in S such that $\Delta(s)(s', a) \neq 0$. A related notion of bounded ambiguity has been utilised to obtain decidability results in the context of probabilistic automata [8]. We introduce the following notation for unambiguous semi-Markov processes: $T(s, w)$ is the state reached after emitting w from s.

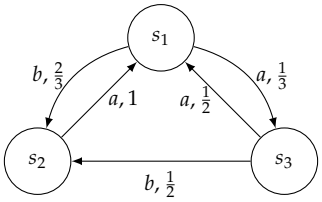


Figure B.5.1: An example of an unambiguous semi-Markov process.

Example B.5.1. Figure B.5.1 gives an example of an unambiguous semi-Markov process. For each of the three states, there is at most one state that can be reached by a given output label. However, there need not be a transition for each output label from every state. In this example, the state s_2 has no b-transition, so for instance $T(s_1, ab) = s_2$, but $T(s_1, abb)$ is undefined. ♦

Theorem B.5.2. *The faster-than problem is decidable in* **coNP** *over unambiguous semi-Markov processes for all effective classes of timing distributions.*

Theorem B.5.2 follows from the next proposition.

Proposition B.5.3. *Consider an unambiguous semi-Markov process $\mathcal{M}$ and two states s, s'. Let $L(s, s')$ be the set of loops reachable from (s, s'):*

$$\left\{ (p, p', v) \in S^2 \times \mathtt{Out}^{\leq S^2} \;\middle|\; \exists w \in \mathtt{Out}^{\leq S^2}, \quad \begin{array}{l} T(s, w) = p,\ T(s', w) = p', \\ T(p, v) = p,\ T(p', v) = p' \end{array} \right\}.$$

We have $s \preceq s'$ if, and only if

- *for all w in $\mathtt{Out}^{\leq S^2}$, we have $\mathbb{P}_{\mathcal{M}}(s, w) \geq \mathbb{P}_{\mathcal{M}}(s', w)$, and*
- *for all (p, p', v) in $L(s, s')$, we have $\mathbb{P}_{\mathcal{M}}(p, v) \geq \mathbb{P}_{\mathcal{M}}(p', v)$.*

Before going into the proof, we explain how to use Proposition B.5.3 to construct an algorithm solving the faster-than problem over unambiguous semi-Markov processes.

1. The first step is to compute $L(s, s')$, which can be done in polynomial time using a simple graph analysis,
2. The second step is to check the two properties, which both can be reduced to exponentially many queries of the form: $\mu_1 \geq \mu_2$ for μ_1, μ_2 in $\mathrm{Conv}(\mathcal{C})$.

To obtain a **coNP** algorithm, in the second step we guess which of the two properties is not satisfied and a witness of polynomial length, which is either a word of quadratic length for the first property, or two states and a word of quadratic length for the second property.

We split the proof of Proposition B.5.3 into two lemmas, each proving one direction of the proposition. The following lemma gives the first direction.

Lemma B.5.4. *If $s \preceq s'$, then, for all $(p, p', v) \in L(s, s')$, $\mathbb{P}_{\mathcal{M}}(p, v) \geq \mathbb{P}_{\mathcal{M}}(p', v)$.*

Proof. Assume that s is faster than s' and let (p, p') be in $L(s, s')$. There exist w, v in $\mathtt{Out}^*$ such that $T(s, w) = p,\ T(s', w) = p',\ T(p, v) = p,\ T(p', v) = p'$. Let n in $\mathbb{N}$. Since s is faster than s', we have $\mathbb{P}_{\mathcal{M}}(s, wv^n) \geq \mathbb{P}_{\mathcal{M}}(s', wv^n)$. We have

$$\mathbb{P}_{\mathcal{M}}(s, wv^n) = \mathbb{P}_{\mathcal{M}}(s, w) * \underbrace{\mathbb{P}_{\mathcal{M}}(p, v) * \cdots * \mathbb{P}_{\mathcal{M}}(p, v)}_{n \text{ times}}$$
$$\mathbb{P}_{\mathcal{M}}(s', wv^n) = \mathbb{P}_{\mathcal{M}}(s', w) * \underbrace{\mathbb{P}_{\mathcal{M}}(p', v) * \cdots * \mathbb{P}_{\mathcal{M}}(p', v)}_{n \text{ times}}.$$

Let $X_{s,w}$ be the random variable measuring the time elapsed from s emitting w. Similarly, we define $X_{p,v}, Y_{s',w}$ and $Y_{p',v}$. We have: for all n in $\mathbb{N}$, for all t,

$$\mathbb{P}_{\mathcal{M}}(X_{s,w} + nX_{p,v} \leq t) \geq \mathbb{P}_{\mathcal{M}}(Y_{s',w} + nY_{p',v} \leq t),$$

Dividing both sides by n yields

$$\mathbb{P}_{\mathcal{M}}\left(\frac{X_{s,w}}{n} + X_{p,v} \leq \frac{t}{n}\right) \geq \mathbb{P}_{\mathcal{M}}\left(\frac{Y_{s',w}}{n} + Y_{p',v} \leq \frac{t}{n}\right).$$

We make the change of variables $x = \frac{t}{n}$: for all n in $\mathbb{N}$, for all x we have

$$\mathbb{P}_{\mathcal{M}}\left(\frac{X_{s,w}}{n} + X_{p,v} \leq x\right) \geq \mathbb{P}_{\mathcal{M}}\left(\frac{Y_{s',w}}{n} + Y_{p',v} \leq x\right).$$

Letting $n \to \infty$, we then obtain, for all x

$$\mathbb{P}_{\mathcal{M}}(X_{p,v} \leq x) \geq \mathbb{P}_{\mathcal{M}}(Y_{p',v} \leq x),$$

which is equivalent to $\mathbb{P}_{\mathcal{M}}(p, v) \geq \mathbb{P}_{\mathcal{M}}(p', v)$. ■

The following lemma gives the converse implication of Proposition B.5.3.

Lemma B.5.5. *Assume that*

- *for all w in $\mathtt{Out}^{\leq S^2}$, we have $\mathbb{P}_{\mathcal{M}}(s, w) \geq \mathbb{P}_{\mathcal{M}}(s', w)$, and*
- *for all (p, p', v) in $L(s, s')$, we have $\mathbb{P}_{\mathcal{M}}(p, v) \geq \mathbb{P}_{\mathcal{M}}(p', v)$.*

Then $s \preceq s'$.

Proof. We prove that for all w, we have $\mathbb{P}_{\mathcal{M}}(s, w) \geq \mathbb{P}_{\mathcal{M}}(s', w)$ by induction on the length of w.

For w of length at most S^2, this is ensured by the first assumption. Let w be a word longer than S^2. There exist two states p, p' such that p is reached by s and p' by s' after emitting i letters of w and again after emitting j letters

of w, with j at most S^2. Let $w = w_1\ v\ w_2$ where v starts at position i and ends at position j. By construction (p, p', v) is in $L(s, s')$. We have

$$\begin{aligned}
\mathbb{P}_{\mathcal{M}}(s, w) &= \mathbb{P}_{\mathcal{M}}(s, w_1) * \mathbb{P}_{\mathcal{M}}(p, v) * \mathbb{P}_{\mathcal{M}}(p, w_2) \\
&= \mathbb{P}_{\mathcal{M}}(s, w_1) * \mathbb{P}_{\mathcal{M}}(p, w_2) * \mathbb{P}_{\mathcal{M}}(p, v) \\
&= \mathbb{P}_{\mathcal{M}}(s, w_1 w_2) * \mathbb{P}_{\mathcal{M}}(p, v) \\
&\geq \mathbb{P}_{\mathcal{M}}(s', w_1 w_2) * \mathbb{P}_{\mathcal{M}}(p', v) \\
&= \mathbb{P}_{\mathcal{M}}(s', w_1) * \mathbb{P}_{\mathcal{M}}(p', w_2) * \mathbb{P}_{\mathcal{M}}(p', v) \\
&= \mathbb{P}_{\mathcal{M}}(s', w_1) * \mathbb{P}_{\mathcal{M}}(p', v) * \mathbb{P}_{\mathcal{M}}(p', w_2) \\
&= \mathbb{P}_{\mathcal{M}}(s', w).
\end{aligned}$$

The equalities use the associativity and commutativity of the convolution. The inequality $\mathbb{P}_{\mathcal{M}}(s, w_1 w_2) \geq \mathbb{P}_{\mathcal{M}}(s', w_1 w_2)$ holds by induction hypothesis, because $w_1 w_2$ is shorter than w. The inequality $\mathbb{P}_{\mathcal{M}}(p, v) \geq \mathbb{P}_{\mathcal{M}}(p', v)$ holds thanks to the second assumption. ■

B.6 Logic

In this section we give a logical characterisation of the faster-than relation. The logic needed for this turns out to be quite simple, and it therefore possesses many nice properties. In particular, every formula is satisfiable by a finite model.

The logic $\mathcal{L}$ consists of path formulas

$$\varphi ::= \top \mid \langle a \rangle \varphi$$

and state formulas

$$\psi ::= \mathcal{P}^{\leq t}_{\geq p}(\varphi)$$

where $t, p \in \mathbb{Q}_{\geq 0}$.

For the semantics of $\mathcal{L}$, we consider paths $\pi = a_1 a_2 \cdots \in \mathtt{Out}^*$ to be infinite sequences of output labels, and we let $\pi[i] = a_i$ be the ith label of π. The semantics are then given by

$$\begin{array}{lll}
\pi \models \top & & \text{always} \\
\pi \models \langle a \rangle \varphi & \text{iff} & \pi[1] = a \text{ and } \pi|_2 \models \varphi \\
\mathcal{M}, s \models \mathcal{P}^{\leq t}_{\geq p}(\varphi) & \text{iff} & \mathbb{P}_{\mathcal{M}}(s, \mathfrak{W}(\varphi))(t) \geq p
\end{array}$$

where $\pi|_2$ is the tail of π, and $\mathfrak{W}(\varphi)$ is the longest common prefix of all paths which satisfy φ.

Theorem B.6.1. *$s \preceq s'$ if and only if $\mathcal{M}, s' \models \psi$ implies $\mathcal{M}, s \models \psi$, for all $\psi \in \mathcal{L}$.*

Proof. ($\Longrightarrow$) Let $s \preceq s'$ and assume $\mathcal{M}, s' \models \mathcal{P}^{\leq t}_{\geq p}(\varphi)$. One can easily prove by structural induction on φ that $\mathbb{P}_{\mathcal{M}}(s', \mathfrak{W}(\varphi)) = \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)$ for some $a_1, \dots, a_n$. Hence we know that

$$\mathbb{P}_{\mathcal{M}}(s', \mathfrak{W}(\varphi))(t) = \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)(t) \geq p,$$

and since $s \preceq s'$, this implies that

$$\mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(t) \geq \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)(t) \geq p,$$

so $\mathbb{P}_{\mathcal{M}}(s, \mathfrak{W}(\varphi))(t) \geq p$.

($\Longleftarrow$) We show the contrapositive. Assume that $s \not\preceq s'$, meaning that there exists $a_1 \dots a_n$ and t such that

$$\mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(t) < \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)(t).$$

Then we can find a rational q such that

$$\mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(t) < q < \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)(t).$$

Now let $\varepsilon = q - \mathbb{P}(s, a_1 \dots a_n)(t) > 0$. By right-continuity, there exists some $\delta > 0$ such that $t < x < t + \delta$ implies

$$\mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(x) - \mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(t) < \varepsilon.$$

Choose a rational q' such that $t < q' < t + \delta$ in order to obtain

$$\mathbb{P}_{\mathcal{M}}(s, a_1 \dots a_n)(q') < q \leq \mathbb{P}_{\mathcal{M}}(s', a_1 \dots a_n)(q').$$

But then we have

$$\mathcal{M}, s' \models \mathcal{P}^{\leq q'}_{\geq q}(\langle a_1 \rangle \dots \langle a_n \rangle \top) \quad \text{and} \quad \mathcal{M}, s \not\models \mathcal{P}^{\leq q'}_{\geq q}(\langle a_1 \rangle \dots \langle a_n \rangle \top). \qquad \blacksquare$$

Next we show that every formula has a finite model, which also implies that every formula is satisfiable.

Theorem B.6.2 (Finite model property)**.** *Any formula $\psi \in \mathcal{L}$ has a finite semi-Markov process satisfying it.*

Proof. For any path formula $\varphi = \langle a_1 \rangle \dots \langle a_n \rangle \top$, we construct the model $\mathcal{M}_\varphi = (S, \tau, \rho)$ as follows. Let $S = \{s_1, \dots s_{n+1}\}$, and let

$$\tau(s_i)(s_{i+1}, a) = \begin{cases} 1 & \text{if } a = a_i \\ 0 & \text{otherwise.} \end{cases}$$

Finally, let $\rho(s) = \delta_0$ be the Dirac distribution at 0 for all states.

Now it is easy to see that $\mathcal{M}_\varphi, s_1 \models \mathcal{P}^{\leq t}_{\geq p}(\varphi)$ for any t and q. $\blacksquare$

Corollary B.6.3. *Every formula ψ is satisfiable, and hence the satisfiability problem is trivially decidable.*

Lastly we consider the model checking problem for $\mathcal{L}$. This problem can be solved by once more making use of the existential theory of the reals, thus giving a **PSPACE** algorithm.

Theorem B.6.4. *The model checking problem is decidable for any semi-Markov process with residence-time distributions from one of the following classes of timing distributions.*

- *Exponential distributions,*
- *piecewise polynomial distributions,*
- *piecewise affine distributions,*
- *uniform distributions, or*
- *Dirac distributions.*

Proof. This essentially follows from Proposition B.2.7, by letting the right-hand side of the inequality be a constant. ■

B.7 Conclusion and Open Problems

We studied the model of semi-Markov processes where the timing behaviour can be described by arbitrary timing distributions. We have introduced a trace-based relation called the faster-than relation which asks that for any prefix and any time bound, the probability of outputting a word with that prefix within the time bound is higher in the faster process than in the slower process. We have shown through a connection to probabilistic automata that the faster-than relation is highly undecidable. It is undecidable in general, and remains Positivity-hard even for one output label. Furthermore, approximating the faster-than relation up to a multiplicative constant is impossible.

However, we constructed algorithms for special cases of the faster-than problem. We have shown that if one considers approximating up to an additive constant rather than a multiplicative constant, and if one gives a bound on the time up to which one is interested in comparing the two processes, then approximation can be done for timing distributions in which we are sure to spend some amount of time to take a transition. In addition, we have shown that the faster-than relation is decidable and in **coNP** for unambiguous processes, in which there is a unique successor state for every output label. Furthermore, we have given a logical characterisation of the faster-than relation and shown that both the satisfiability and the model checking problem for this logic are decidable.

In this paper, we have focused on the generative model, where the labels are treated as outputs. An alternative viewpoint is the reactive model, where the labels are instead treated as inputs [19]. While all the undecidability and hardness results we have shown can also easily be shown to hold for the reactive case, the same is not true for the algorithms we have constructed. It is non-trivial to extend these algorithms to the reactive case, and the main obstacle in doing so is that for reactive systems, one has to also handle schedulers, often uncountably many.

B.8 References

[1] S. Akshay, T. Antonopoulos, J. Ouaknine, and J. Worrell, "Reachability problems for Markov chains," *Inf. Process. Lett.*, vol. 115, no. 2, pp. 155–158, 2015.

[2] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[3] C. Baier, J. Katoen, H. Hermanns, and V. Wolf, "Comparative branching-time semantics for Markov chains," *Inf. Comput.*, vol. 200, no. 2, pp. 149–214, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2005.03.001

[4] J.-L. Bon and E. Păltănea, "Ordering properties of convolutions of exponential random variables," *Lifetime Data Analysis*, vol. 5, no. 2, pp. 185–192, 1999.

[5] J. F. Canny, "Some algebraic and geometric computations in PSPACE," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, J. Simon, Ed. ACM, 1988, pp. 460–467. [Online]. Available: http://doi.acm.org/10.1145/62212.62257

[6] A. Condon and R. J. Lipton, "On the complexity of space bounded interactive proofs (extended abstract)," in *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*. IEEE Computer Society, 1989, pp. 462–467. [Online]. Available: https://doi.org/10.1109/SFCS.1989.63519

[7] N. Fijalkow, "Undecidability results for probabilistic automata," *SIGLOG News*, vol. 4, no. 4, pp. 10–17, 2017. [Online]. Available: http://doi.acm.org/10.1145/3157831.3157833

[8] N. Fijalkow, C. Riveros, and J. Worrell, "Probabilistic automata of bounded ambiguity," in *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, ser. LIPIcs, R. Meyer and U. Nestmann, Eds., vol. 85. Schloss Dagstuhl

- Leibniz-Zentrum fuer Informatik, 2017, pp. 19:1–19:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.CONCUR.2017.19

[9] H. Gimbert and Y. Oualhadj, "Probabilistic automata on finite words: Decidable and undecidable problems," in *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, ser. Lecture Notes in Computer Science, S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, Eds., vol. 6199. Springer, 2010, pp. 527–538. [Online]. Available: https://doi.org/10.1007/978-3-642-14162-1_44

[10] S. Guha, C. Narayan, and S. Arun-Kumar, "On decidability of prebisimulation for timed automata," in *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings*, ser. Lecture Notes in Computer Science, P. Madhusudan and S. A. Seshia, Eds., vol. 7358. Springer, 2012, pp. 444–461. [Online]. Available: https://doi.org/10.1007/978-3-642-31424-7_33

[11] F. Killmann and E. von Collani, "A note on the convolution of the uniform and related distributions and their use in quality control," *Economic Quality Control*, vol. 16, no. 1, pp. 17–41, 2010.

[12] G. Lüttgen and W. Vogler, "A faster-than relation for asynchronous processes," in *CONCUR 2001 - Concurrency Theory, 12th International Conference, Aalborg, Denmark, August 20-25, 2001, Proceedings*, ser. Lecture Notes in Computer Science, K. G. Larsen and M. Nielsen, Eds., vol. 2154. Springer, 2001, pp. 262–276. [Online]. Available: https://doi.org/10.1007/3-540-44685-0_18

[13] F. Moller and C. M. N. Tofts, "Relating processes with respect to speed," in *CONCUR '91, 2nd International Conference on Concurrency Theory, Amsterdam, The Netherlands, August 26-29, 1991, Proceedings*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. F. Groote, Eds., vol. 527. Springer, 1991, pp. 424–438. [Online]. Available: https://doi.org/10.1007/3-540-54430-5_104

[14] J. Ouaknine and J. Worrell, "Positivity problems for low-order linear recurrence sequences," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, C. Chekuri, Ed. SIAM, 2014, pp. 366–379. [Online]. Available: https://doi.org/10.1137/1.9781611973402.27

[15] A. Paz, *Introduction to Probabilistic Automata*. Academic Press, 1971.

[16] M. Perman, A. Senegacnik, and M. Tuma, "Semi-Markov models with an application to power-plant reliability analysis," *IEEE Transactions on Reliability*, vol. 46, no. 4, pp. 526–532, Dec 1997.

[17] A. Pievatolo, E. Tironi, and I. Valade, "Semi-Markov processes for power system reliability assessment with application to uninterruptible power supply," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1326–1333, Aug 2004.

[18] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*. University of California press, 1951.

[19] R. J. van Glabbeek, S. A. Smolka, and B. Steffen, "Reactive, generative and stratified models of probabilistic processes," *Inf. Comput.*, vol. 121, no. 1, pp. 59–80, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1123

Paper C

A Faster-Than Relation for Semi-Markov Decision Processes

Mathias R. Pedersen, Giorgio Bacci, and Kim G. Larsen

The paper is based on an unpublished manuscript.

Abstract

When modeling concurrent or cyber-physical systems, non-functional requirements such as time are important to consider. In order to improve the timing aspects of a model, it is necessary to have some notion of what it means for a process to be faster than another, which can guide the stepwise refinement of the model. To this end we study a faster-than relation *for semi-Markov decision processes and compare it to standard notions for relating systems. We show that checking whether a system is faster than another one is undecidable, but as a positive result we give a decision procedure for approximating it. Furthermore, we consider the compositional aspects of this relation, and show that the faster-than relation is not a precongruence with respect to parallel composition, hence giving rise to so-called parallel timing anomalies. We take the first steps toward understanding this problem by identifying decidable conditions sufficient to avoid parallel timing anomalies in the absence of non-determinism.*

C.1 Introduction

Timing aspects are important when considering real-time or cyber-physical systems. For example, they are of interest in real-time embedded systems when one wants to verify the worst-case execution time for guaranteeing minimal system performance or in safety-critical systems when one needs to ensure that unavoidable rigid deadlines will always be met [12].

Semi-Markov decision processes are continuous-time Markov decision processes where the residence-time on states is governed by generic distributions on the positive real line. These systems have been extensively used to model real-time cyber-physical systems [16, 26].

For reasoning about timing aspects it is important to understand what it formally means for a real-time or cyber-physical system to operate faster than another. To this end we define the notion of *faster-than relation* for semi-Markov decision processes. The definition of faster-than relation we propose in this paper is a reactive version of an analogous notion of faster-than relation previously introduced in [19] for the case of generative systems. According to our relation, a semi-Markov decision process is faster than another one when it reacts to any sequence of inputs with equal or higher probability than the slower process, within the same time bound.

Similarly to [19], we show that also the faster-than relation on semi-Markov decision processes is undecidable. However, by extending the approximation algorithm from [19], we obtain an approximation algorithm for the case where we only consider timed events within some fixed time bound. The extension of the algorithm in [19] is not a trivial task, because the definition of faster-than relation on semi-Markov decision processes requires us

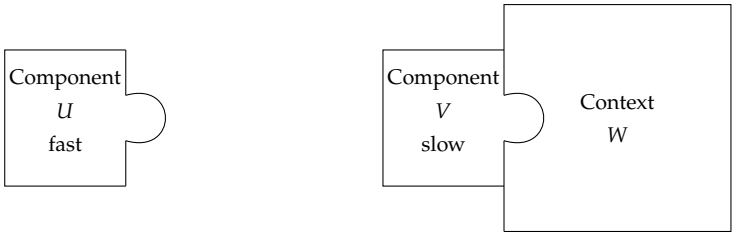


Figure C.1.1: The context W operates in parallel with the component V. If the component U is faster than V, then if we replace V with U, we would expect the overall behaviour to also be faster.

to deal with universal and existential quantifications over schedulers, which were not present in the original definition in [19] for the case of generative systems.

Often, complex cyber-physical systems are organised as concurrent systems of multiple components running in parallel and interacting with each other. Such systems are better analysed *compositionally*, that is, by breaking them into smaller components that are more easily examined [4]. However, it is not always the case that an analysis on the components carries over to the full composite system. A well known example of this, occurring in real-time systems such as scheduling for processors [3, 13], are *timing anomalies*, that is, when locally faster behaviour leads to a globally slower behaviour [11].

In this paper we study the compositional aspects of the faster-than relation for semi-Markov decision processes. The situation we are interested in is depicted in Figure C.1.1 where we have a composite system consisting of a context W and a component V, and we want to understand what happens when we replace V with another component U that is faster than V. We consider some common notions of parallel composition, and show that timing anomalies can occur using our faster-than relation, even in the absence of non-determinism. This shows that timing anomalies are not caused by non-determinism, but arise from the linear timing behaviour of processes.

We then take a first step toward recovering compositional reasoning for the faster-than relation, by identifying conditions sufficient for avoiding timing anomalies, which we call *monotonicity*. Presently we do not know whether these conditions are decidable, however we introduce another set of conditions, called *strong monotonicity*, which are decidable. Unfortunately, strong monotonicity only applies to processes which have no non-determinism.

Related Work.

The notion of a faster-than relation has been studied in many different contexts throughout the literature. The work most closely related to ours is that

of Pedersen et al. [19], which considers a generative version of the faster-than relation, whereas we study the reactive version. The focus of [19] is on decidability issues, and the faster-than relation is proved undecidable. However, positive results are also given in the form of an approximation algorithm, and a decidability result for unambiguous processes. Baier et al. [1] define, among other relations, a simulation relation for continuous-time Markov chains which can be interpreted as a faster-than relation, and study its logical characterisation. However, none of these works consider compositional aspects.

For process algebras, discrete-time faster-than relations have been defined for variations of Milner's CCS, and shown to be precongruences with respect to parallel composition [5, 14, 17, 22]. Lüttgen and Vogler [15] attempt to unify some of these process algebraic approaches and also consider the issue of parallel timing anomalies. For Petri nets, Vogler [27, 28] considers a testing preorder as a faster-than relation and shows that this is a precongruence with respect to parallel composition. Geilen et al. [6] introduces a refinement principle for timed actor interfaces under the slogan "the earlier, the better", which can also be seen as an example of a faster-than relation.

Work on timing anomalies date back to at least 1969 [8], but the most influential paper in the area is probably that of Lundqvist and Stenström [13], in which they show that timing anomalies can occur in dynamically scheduled processors, contrary to what most people assumed at the time. More recent work has focused on compositional aspects [11] and defining timing anomalies formally, using transition systems as the formalism [3, 21].

C.2 Notation and Preliminaries

In this section we fix some notation and recall concepts that are used throughout the rest of the paper. Let $\mathbb{N}$ denote the natural numbers and let $\mathbb{R}_{\geq 0}$ denote the non-negative real numbers, which we equip with the standard Borel σ-algebra $\mathbb{B}$. For any set X, let $\mathcal{D}(X)$ denote the set of probability measures on X, and let $\mathcal{D}_{\leq}(X)$ denote the set of subprobability measures on X. For an element $x \in X$ of some set X, we will use δ_x to denote the Dirac measure at x defined as $\delta_x(y) = 1$ if $x = y$ and $\delta_x(y) = 0$ otherwise. We fix a non-empty, countable set L of *labels* or *actions* and equip them with the discrete σ-algebra Σ_L.

For a probability measure $\mu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, we denote by F_μ its *cumulative distribution function (CDF)* defined as $F_\mu(t) = \mu([0,t])$, for all $t \in \mathbb{R}_{\geq 0}$. We will denote by $Exp[\theta]$ the CDF of an exponential distribution with rate $\theta > 0$. The *convolution* of two probability measures $\mu, \nu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, written $\mu * \nu$, is the probability measure on $\mathbb{R}_{\geq 0}$ given by $(\mu * \nu)(B) = \int_{-\infty}^{\infty} \nu(B - x)\ \mu(\mathrm{d}x)$, for all $B \in \mathbb{B}$ [2]. Convolution is associative, i.e., $\mu * (\nu * \eta) = (\mu * \nu) * \eta$, and

commutative, i.e., $\mu * \nu = \nu * \mu$.

C.3 Semi-Markov Decision Processes

In this section we recall the definition of semi-Markov decision processes.

Definition C.3.1. A *semi-Markov decision process (SMDP)* is a tuple $M = (S, \tau, \rho)$ where

- S is a non-empty, countable set of *states*,
- $\tau : S \times L \to \mathcal{D}_{\leq}(S)$ is a *transition probability function*, and
- $\rho : S \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is a *residence-time probability function*. ▲

The operational behaviour of an SMDP $M = (S, \tau, \rho)$ is as follows. The process in the state $s \in S$ reacts to an external input $a \in L$ provided by the environment by changing its state to $s' \in S$ within time $t \in \mathbb{R}_{\geq 0}$ with probability $\tau(s, a)(s') \cdot \rho(s)([0, t])$.

Notice that Markov decision processes are a special case of SMDPs where for all $s \in S$, $\rho(s) = \delta_0$ (i.e. transitions happen instantaneously), and that continuous-time Markov decision processes are also a special case of SMDPs where, for all states $s \in S$, $F_{\rho(s)} = Exp[\theta_s]$ for some rate $\theta_s \in \mathbb{R}_{\geq 0}$.

The executions of an SMDP $M = (S, \tau, \rho)$ are infinite timed transition sequences of the form $\pi = (s_1, t_1, a_1)(s_2, t_2, a_2) \cdots \in (S \times \mathbb{R}_{\geq 0} \times L)^{\omega}$, representing the fact that M waited in state s_i for t_i time units after the action a_i was input. We will refer to executions of an SMDP as *timed action paths*. For $i \in \mathbb{N}$, let $\pi[i] = s_i$, $\pi\langle i \rangle = t_i$, $\pi[\![i]\!] = a_i$, $\pi|_i = (s_1, t_1, a_1) \dots (s_i, t_i, a_i)$, and $\pi|^i = (s_i, t_i, a_i)(s_{i+1}, t_{i+1}, a_{i+1}) \dots$. We let $\Pi(M)$ denote the set of all timed action paths in M, and denote by $\Pi_n(M) = \{\pi|_n \mid \pi \in \Pi(M)\}$ the set of all prefixes of length n. Hereafter, we refer to timed action paths simply as paths, unless we wish to distinguish between different kinds of paths.

Next we recall the standard construction of the measurable space of paths. A *cylinder set* of rank $n \geq 1$ is the set of all paths whose nth prefix is contained in a common subset $E \subseteq \Pi_n(M)$, and is given by

$$\mathfrak{C}(E) = \{\pi \in \Pi(M) \mid \pi|_n \in E\}.$$

It will be convenient to denote *rectangular cylinders* of the form

$$\mathfrak{C}(S_1 \times L_1 \times R_1 \times \cdots \times S_n \times L_n \times R_n),$$

for $S_i \subseteq S$, $L_i \subseteq L$, and $R_i \subseteq \mathbb{R}_{\geq 0}$, as

$$\mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n).$$

We denote by $(\Pi(M), \Sigma)$ the *measurable space of timed action paths*, where Σ is the smallest σ-algebra generated by the cylinders of the form

$$\mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n)$$

for $S_i \in 2^S$, $L_i \in 2^L$, and $R_i \in \mathbb{B}$.

In this paper we assume that external choices are resolved by means of memoryless stochastic schedulers, however all the results we present still hold for memoryful schedulers.

Definition C.3.2. Given an SMDP $M = (S, \tau, \rho)$, a *scheduler* for M is a function $\sigma : S \to \mathcal{D}(L)$ that assigns to each state a probability distribution over action labels. ▲

We will use the notation $\tau^\sigma(s,a)(s')$ as shorthand for $\tau(s,a)(s') \cdot \sigma(s)(a)$ to denote the probability of moving from state s to s' under the stochastic choice of a given by σ. Given an SMDP M and a scheduler σ for it, the probabilistic execution of a path starting from the state s is governed by the probability $\mathbb{P}^\sigma_M(s)$ on $(\Pi(M), \Sigma)$ defined as follows.

Definition C.3.3. Let $M = (S, \tau, \rho)$ be an SMDP. Given a scheduler σ for M and a state $s \in S$, $\mathbb{P}^\sigma_M(s)$ is defined as the unique (sub)probability measure[1] on $(\Pi(M), \Sigma)$ such that for all $S_i \in 2^S$, $L_i \in 2^L$, and $R_i \in \mathbb{B}$, with $1 \leq i \leq n$, we have

$$\mathbb{P}^\sigma_M(s)(\mathfrak{C}(S_1, L_1, R_1)) = \rho(s)(R_1) \cdot \sum_{a \in L_1} \sum_{s' \in S_1} \tau^\sigma(s,a)(s')$$

and

$$\begin{aligned} &\mathbb{P}^\sigma_M(s)(\mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n)) \\ &= \rho(s)(R_1) \cdot \sum_{a \in L_1} \sum_{s' \in S_1} \tau^\sigma(s,a)(s') \cdot \mathbb{P}^\sigma_M(s')(\mathfrak{C}(S_2 \dots S_n, L_2 \dots L_n, R_2 \dots R_n)). \end{aligned}$$

▲

Intuitively, to get the probability $\mathbb{P}^\sigma_M(s)(\mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n))$, we first take the probability that s takes a transition at a time point in R_1, given by $\rho(s)(R_1)$, after which we sum over the probabilities of all the possible transitions that can be taken by choosing a label $a \in L_1$ and a state $s' \in S_1$, and then the rest of the probability is given inductively by continuing on s'. For the rest of the paper, we will omit the subscript M in $\mathbb{P}^\sigma_M$ whenever it is clear from the context which SMDP is being referred to.

[1] Existence and uniqueness is guaranteed by the Hahn-Kolmogorov theorem [24].

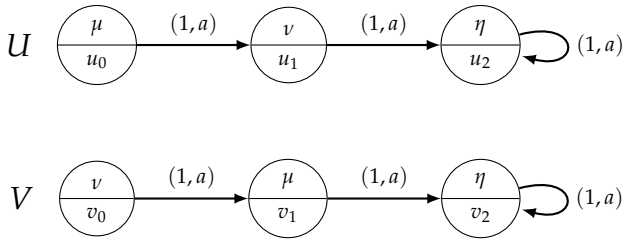


Figure C.4.1: If $F_\mu(t) \geq F_\nu(t)$ for all t, then U is faster than V in the first states, and after that their probabilities are the same, so U is faster than V.

C.4 A Faster-Than Relation

Our aim is to define a relation that formalises the intuitive idea of an SMDP U being "faster than" another SMDP V. For a process U to be faster than V, it must be able to execute any sequence of actions $a_1, \ldots, a_n$ in less time than V. Since we are dealing with probabilistic systems, we must speak of the probability of executing a sequence of actions within some time bound.

Consider the two simple SMDPs U and V in Figure C.4.1 with just a single transition label and initial states u_0 and v_0, respectively. Here μ, ν, η are arbitrary probability measures on $\mathbb{R}_{\geq 0}$, representing the residence-time distributions at each state. An arrow with label (p, a) means that when a is chosen as the action, then the SMDP takes the transition given by the arrow with probability p. The only finite sequences of actions that can be executed in these SMDPs are of the form a^n for $n > 0$.

For U to be faster than V, it should be the case that for any time bound t and no matter which scheduler σ we choose for V, we must be able to find a scheduler σ' for U such that there is an earlier time bound $t' \leq t$ which allows U to execute any sequence a^n within time t' with higher or equal probability than that of V executing the same sequence of actions within time t. Formally, this amounts to saying that $\mathbb{P}^{\sigma'}(u_0)(\mathfrak{C}(a^n, t')) \geq \mathbb{P}^{\sigma}(v_0)(\mathfrak{C}(a^n, t))$, where $\mathfrak{C}(a_1 \ldots a_n, t)$ denotes the event of executing the sequence of actions $a_1, \ldots, a_n$ within time t. Hence, the type of events on which we want to focus are the following.

Definition C.4.1. For any finite sequence of actions $a_1, \ldots, a_n$, and $t \in \mathbb{R}_{\geq 0}$, we say that

$$\mathfrak{C}(a_1 \ldots a_n, t) = \{\pi \in \Pi(M) \mid \forall 1 \leq i \leq n,\ \pi[\![i]\!] = a_i \text{ and } \sum_{j=1}^{n} \pi\langle j \rangle \leq t\}$$

is a *time-bounded cylinder*. The *length* of a time-bounded cylinder is the length of the sequence of actions in the time-bounded cylinder. ▲

Note that $\mathfrak{C}(a_1 \dots a_n, t)$ is measurable in $(\Pi(M), \Sigma)$, since

$$f : \Pi_n(\mathcal{M}) \to S^n \times L^n \times \mathbb{R}^n_{\geq 0}$$

given by

$$f((s_1, o_1, t_1), \dots, (s_n, o_n, t_n)) = (s_1, \dots, s_n), (o_1, \dots, o_n), (t_1, \dots, t_n)$$

and

$$\mathtt{res}_n : \Pi(\mathcal{M}) \to \Pi_n(\mathcal{M})$$

given by

$$\mathtt{res}_n(\pi) = \pi|_n$$

are both measurable, and hence

$$(f \circ \mathtt{res}_n)^{-1}(S^n \times \{(a_1, \dots, a_n)\} \times B^n_t) = \mathfrak{C}(a_1 \dots a_n, t)$$

is measurable, where $B^n_t = \{(r_1, \dots, r_n) \in \mathbb{R}^n_{\geq 0} \mid \sum_{i=1}^n r_i \leq t\}$.

Example C.4.2. The time-bounded cylinder $\mathfrak{C}(aa, 2)$ denotes the set of all paths where the first two output labels are both a's, and the first two steps of the path are completed within 2 time units. ⧫

We will use the notation (M, s_0) to indicate that $M = (S, \tau, \rho)$ is an SMDP with initial state $s_0 \in S$ and call it *pointed SMDP*. For the rest of the paper, we fix three SMDPs $M = (S, \tau, \rho)$, $U = (S_U, \tau_U, \rho_U)$, and $V = (S_V, \tau_V, \rho_V)$, with initial states $s_0 \in S$, $u_0 \in S_U$, $v_0 \in S_V$, respectively. Now we are ready to define what it means for an SMDP to be "faster than" another one.

Definition C.4.3 (Faster-than)**.** We say that U is *faster than* V, written $U \preceq V$, if for all schedulers σ for V, time bounds t, and sequences of actions $a_1 \dots a_n$, there exists a scheduler σ' for U and time bound $t' \leq t$, such that $\mathbb{P}^{\sigma'}(u_0)(\mathfrak{C}(a_1 \dots a_n, t')) \geq \mathbb{P}^{\sigma}(v_0)(\mathfrak{C}(a_1 \dots a_n, t))$. ▲

Clearly, the faster-than relation $\preceq$ is a preorder. The following proposition gives a characterisation of the faster-than relation that is often easier to work with.

Proposition C.4.4. *$U \preceq V$ if and only if for all schedulers σ for V there exists a scheduler σ' for U such that $\mathbb{P}^{\sigma'}(u_0)(C) \geq \mathbb{P}^{\sigma}(v_0)(C)$, for all time-bounded cylinders C.*

Proof. Clearly, if for all schedulers σ for V there exists a scheduler σ' for U such that $\mathbb{P}^{\sigma}(u_0)(C) \geq \mathbb{P}^{\sigma'}(v_0)(C)$ for all time-bounded cylinders C, then $U \preceq V$ by taking $C' = C$. If $U \preceq V$, then consider an arbitrary scheduler

σ, and time-bounded cylinder $C = \mathfrak{C}(a_1 \dots a_n, t)$. There exists a scheduler σ' and $t' \in \mathbb{R}_{\geq 0}$ such that $t \geq t'$ and

$$\mathbb{P}^{\sigma'}(u_0)(a_1 \dots a_n, t') \geq \mathbb{P}^{\sigma}(v_0)(a_1 \dots a_n, t).$$

By monotonicity, $t \geq t'$ implies that

$$\mathbb{P}^{\sigma'}(u_0)(a_1 \dots a_n, t) \geq \mathbb{P}^{\sigma'}(u_0)(a_1 \dots a_n, t'),$$

and hence $\mathbb{P}^{\sigma'}(u_0)(C) \geq \mathbb{P}^{\sigma}(v_0)(C)$. ■

Before showing an example of an SMDP being faster than another one, we provide an analytic solution for computing the probability over time-bounded cylinders in terms of convolutions of the residence time distributions.

Proposition C.4.5. *For any SMDP M, scheduler σ for M, and $s \in S$, we have*

$$\begin{aligned}
&\mathbb{P}^{\sigma}(s)(\mathfrak{C}(S_1 \dots S_n, L_1 \dots L_n, R_1 \dots R_n)) \\
&= \sum_{s_n \in S_n} \sum_{a_n \in L_n} \cdots \sum_{s_1 \in S_1} \sum_{a_1 \in L_1} \tau^{\sigma}(s, a_1)(s_1) \cdots \tau^{\sigma}(s_{n-1}, a_n)(s_n) \\
&\quad \cdot \rho(s) \times \rho(s_1) \times \cdots \times \rho(s_{n-1})(R_1 \times R_2 \times \cdots \times R_n).
\end{aligned}$$

Proof. The proof is by induction on the length n of the cylinder. If the cylinder has length $n = 1$ then

$$\mathbb{P}^{\sigma}(s)(\mathfrak{C}(S_1, L_1, R_1)) = \sum_{s_1 \in S_1} \sum_{a_1 \in L_1} \tau^{\sigma}(s, a_1)(s_1) \cdot \rho(s)(R_1).$$

If the cylinder has length $n = k + 1$, then

$$\begin{aligned}
&\mathbb{P}^{\sigma}(s)(\mathfrak{C}(S_1 \dots S_{k+1}, L_1 \dots L_{k+1}, R_1 \dots R_{k+1})) \\
&= \rho(s)(R_1) \cdot \sum_{s_1 \in S_1} \sum_{a_1 \in L_1} \tau^{\sigma}(s, a_1)(s_1) \cdot \\
&\quad \mathbb{P}^{\sigma}(s_1)(\mathfrak{C}(S_2 \dots S_{k+1}, L_2 \dots L_{k+1}, R_2 \dots R_{k+1})) \\
&= \rho(s)(R_1) \cdot \sum_{s_1 \in S_1} \sum_{a_1 \in L_1} \tau^{\sigma}(s, a_1)(s_1) \\
&\quad \cdot \sum_{s_{k+1} \in S_{k+1}} \sum_{a_{k+1} \in L_{k+1}} \cdots \sum_{s_2 \in S_2} \sum_{a_2 \in L_2} \tau^{\sigma}(s_1, a_2)(s_2) \cdots \tau^{\sigma}(s_k, a_{k+1})(s_{k+1}) \\
&\quad \cdot \rho(s_1) \times \cdots \times \rho(s_{k+1})(R_2 \times \cdots \times R_{k+1}) \\
&= \sum_{s_{k+1} \in S_{k+1}} \sum_{a_{k+1} \in L_{k+1}} \cdots \sum_{s_1 \in S_1} \sum_{a_1 \in L_1} \tau^{\sigma}(s, a_1)(s_1) \cdots \tau^{\sigma}(s_k, a_{k+1})(s_{k+1}) \\
&\quad \cdot \rho(s) \times \rho(s_1) \times \cdots \times \rho(s_{k+1})(R_1 \times \cdots \times R_{k+1}).
\end{aligned}$$

■

Corollary C.4.6. *For any SMDP M, scheduler σ for M, $s \in S$, and Borel set $B \in \mathbb{R}^n_{\geq 0}$ we have*

$$\begin{aligned}&\mathbb{P}^\sigma(s)(\mathfrak{C}(S \dots S, \{a_1\} \dots \{a_n\}, B))\\ =& \sum_{s_n \in S} \cdots \sum_{s_1 \in S} \tau^\sigma(s, a_1)(s_1) \cdots \tau^\sigma(s_{n-1}, a_n)(s_n)\\ &\cdot \rho(s) \times \rho(s_1) \times \cdots \times \rho(s_{n-1})(B).\end{aligned}$$

Proposition C.4.7. *For any SMDP $M = (S, \tau, \rho)$, scheduler σ for M, and $s \in S$ we have*

$$\begin{aligned}&\mathbb{P}^\sigma(s)(\mathfrak{C}(a_1 \dots a_n, t))\\ =& \sum_{s_1 \in S} \cdots \sum_{s_n \in S} \tau^\sigma(s, a_1)(s_1) \cdots \tau^\sigma(s_{n-1}, a_n)(s_n)\\ &\cdot (\rho(s) * \rho(s_1) * \cdots * \rho(s_{n-1}))([0, t]).\end{aligned}$$

Proof. By Corollary C.4.6, we know that

$$\begin{aligned}&\mathbb{P}^\sigma(s)(\mathfrak{C}(a_1 \dots a_n, t))\\ =& \sum_{s_n \in S} \cdots \sum_{s_1 \in S} \tau^\sigma(s, a_1)(s_1) \cdots \tau^\sigma(s_{n-1}, a_n)(s_n)\\ &\cdot \rho(s) \times \rho(s_1) \times \cdots \times \rho(s_{n-1})(B^n_t).\end{aligned}$$

Hence, if we can show that

$$\rho(s) \times \rho(s_1) \times \cdots \times \rho(s_{n-1})(B^n_t) = (\rho(s) * \rho(s_1) * \cdots * \rho(s_{n-1}))([0, t]),$$

the proof is done.

The proof now proceeds by induction on the length n of the time-bounded cylinder $\mathfrak{C}(a_1 \dots a_n, t)$. If $n = 1$, then

$$\rho(s)(B^1_t) = \rho(s)([0, t]).$$

If $n = k + 1$, then

$$\begin{aligned}&(\rho(s) \times \rho(s_1) \times \cdots \times \rho(s_k))(B^{k+1}_t)\\ =& \int_0^t (\rho(s_1) \times \cdots \times \rho(s_k))(B^k_{t-x})\ \rho(s)(\mathrm{d}x) && \text{(Fubini)}\\ =& \int_0^t (\rho(s_1) * \cdots * \rho(s_k))([0, t-x])\ \rho(s)(\mathrm{d}x) && \text{(ind. hyp.)}\\ =& (\rho(s) * (\rho(s_1) * \cdots * \rho(s_k)))([0, t]) && \text{(def. of convolution)}\\ =& (\rho(s) * \rho(s_1) * \cdots * \rho(s_k))([0, t]). && \text{(associativity)}\end{aligned}$$

■

Proposition C.4.7 intuitively says that the absorption-time of any path of length n through the SMDP is distributed as the n-fold convolution of its residence-time probabilities. Therefore, the probability of doing transitions with labels $a_1, \ldots, a_n$ within time t is the sum of the probabilities of taking a path of length n with labels $a_1, \ldots, a_n$ through the SMDP, weighted by the probability of reaching the end of each of these paths within time t. This is similar in spirit to a result on phase-type distributions, see e.g. [20, Proposition 2.11].

From Proposition C.4.7 we can also derive the following which gives a more direct inductive definition of the probability on time-bounded cylinders. If we fix $a_1 \ldots a_n$ and let t vary, we get a CDF

$$\mathbb{P}^\sigma(s)(a_1 \ldots a_n)([0,t]) = \mathbb{P}^\sigma(s)(\mathfrak{C}(a_1 \ldots a_n, t)).$$

Proposition C.4.8. *The CDF $\mathbb{P}^\sigma(s)(a_1 \ldots a_n)$ can be characterised inductively by*

$$\mathbb{P}^\sigma(s)(a_1)([0,t]) = \sum_{s' \in S} \tau^\sigma(s, a_1)(s') \cdot \rho(s)([0,t]),$$

$$\mathbb{P}^\sigma(s)(a_1 \ldots a_n)([0,t]) = \sum_{s' \in S} \tau^\sigma(s, a_1)(s') \cdot (\rho(s) * \mathbb{P}^\sigma(s')(a_2 \ldots a_n))([0,t]).$$

Proof. For $n = 1$ we have

$$\mathbb{P}^\sigma(s)(a)([0,t]) = \mathbb{P}^\sigma(s)(\mathfrak{C}(a,t)) = \sum_{s' \in S} \tau^\sigma(s,a)(s') \cdot \rho(s)([0,t]).$$

For $n = k + 1$ we have

$$\begin{aligned}
&\mathbb{P}^\sigma(s)(a_1 \ldots a_n) \\
&= \sum_{s_1 \in S} \cdots \sum_{s_n \in S} \tau^\sigma(s, a_1)(s_1) \cdots \tau^\sigma(s_k, a_n)(s_n) \cdot (\rho(s) * \cdots * \rho(s_n))([0,t]) \\
&= \sum_{s_1 \in S} \cdots \sum_{s_n \in S} \tau^\sigma(s, a_1)(s_1) \cdots \tau^\sigma(s_k, a_n)(s_n) \\
&\quad \cdot \int_0^t (\rho(s_1) * \cdots * \rho(s_n))(t - x)\, \rho(s)(\mathrm{d}x) \\
&= \sum_{s_1 \in S} \tau^\sigma(s, a_1)(s_1) \cdot (\rho(s) * \mathbb{P}^\sigma(s_1)(a_2 \ldots a_n))([0,t]).
\end{aligned}$$

■

Proposition C.4.8 also shows that our definition of faster-than coincides with the one from [19], except ours is reactive rather than generative.

Example C.4.9. Consider the pointed SMDPs (U, u_0) and (V, v_0) that are depicted in Figure C.4.1. Assuming that $F_\mu(t) \geq F_\nu(t)$ for all t, we now show that $U \preceq V$. To compare U and V, first notice that we only need to consider time-bounded cylinders of the form $\mathfrak{C}(a^n, t)$, for $n \geq 1$. Since the set of actions

is $L = \{a\}$, the only possible valid scheduler σ for both U and V is the one assigning the Dirac measure δ_a to all states. We consider two cases.

(Case $n = 1$) In this case we get

$$\mathbb{P}^\sigma(u_0)(\mathfrak{C}(a,t)) = F_\mu(t) \quad \text{and} \quad \mathbb{P}^\sigma(v_0)(\mathfrak{C}(a,t)) = F_\nu(t).$$

Since we assumed $F_\mu(t) \geq F_\nu(t)$ for all t, this implies

$$\mathbb{P}^\sigma(u_0)(\mathfrak{C}(a,t)) \geq \mathbb{P}^\sigma(v_0)(\mathfrak{C}(a,t)).$$

(Case $n > 1$) By Proposition C.4.7 we have both

$$\mathbb{P}^\sigma(u_0)(\mathfrak{C}(a^n,t)) = (\mu * \nu * \eta^{*(n-2)})([0,t])$$

and

$$\mathbb{P}^\sigma(v_0)(\mathfrak{C}(a^n,t)) = (\nu * \mu * \eta^{*(n-2)})([0,t]),$$

where η^{*n} is the n-fold convolution of η, defined inductively by $\eta^{*0} = \delta_0$ and $\eta^{*(n+1)} = \eta * \eta^{*n}$. Since convolution is commutative and associative, and δ_0 is the identity for convolution, we obtain

$$\mathbb{P}^\sigma(u_0)(\mathfrak{C}(a^n,t)) = \mathbb{P}^\sigma(v_0)(\mathfrak{C}(a^n,t)).$$

We therefore conclude that $U \preceq V$. ♦

C.4.1 Comparison With Simulation and Bisimulation

The standard notions used to compare processes are bisimulation [18] and simulation [1]. We next recall their definitions, naturally extended to our setting of SMDPs.

Definition C.4.10. For an SMDP M, a relation $R \subseteq S \times S$ is a *bisimulation relation* (resp. *simulation relation*) on M if for all $(s_1, s_2) \in R$ we have

- $F_{\rho(s_1)}(t) = F_{\rho(s_2)}(t)$ (resp. $F_{\rho(s_1)}(t) \leq F_{\rho(s_2)}(t)$) for all $t \in \mathbb{R}_{\geq 0}$ and
- for all $a \in L$ there exists a weight function $\Delta_a : S \times S \to [0,1]$ such that
  - $\Delta_a(s,s') > 0$ implies $(s,s') \in R$,
  - $\tau(s_1,a)(s) = \sum_{s' \in S} \Delta_a(s,s')$ for all $s \in S$, and
  - $\tau(s_2,a)(s') = \sum_{s \in S} \Delta_a(s,s')$ for all $s' \in S$.

If there is a bisimulation relation (resp. simulation relation) R such that $(s_1, s_2) \in R$, then we say that s_1 and s_2 are *bisimilar* (resp. s_2 *simulates* s_1) and write $s_1 \sim s_2$ (resp. $s_1 \precsim s_2$). ▲

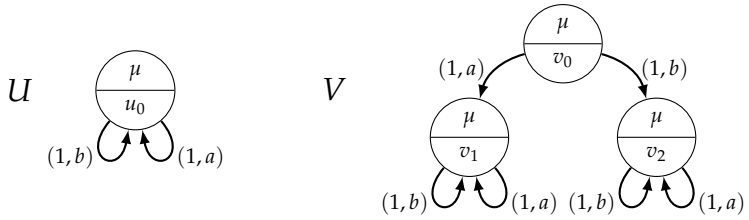


Figure C.4.2: Example showing that the faster-than relation and the simulation relation are incomparable.

We lift bisimulation and simulation relations to two different SMDPs by considering the disjoint union of the two and comparing their initial states. We denote by $\sim$ the largest bisimulation relation and by $\precsim$ the largest simulation relation. Furthermore, we say that U and V are *equally fast* and write $U \equiv V$ if $U \precsim V$ and $V \precsim U$.

Example C.4.11. Consider the two SMDPs U and V in Figure C.4.2 with the same probability measure μ in all states. It is easy to see that U is bisimilar to V, and hence V also simulates U. However, we show that $U \not\precsim V$ in the following way. Construct the scheduler σ for V by letting

$$\sigma(v_0)(a) = 0.5,\ \sigma(v_0)(b) = 0.5,\ \sigma(v_1)(a) = 1,\ \text{and}\ \sigma(v_2)(b) = 1.$$

Now, for any scheduler σ' for U, we must have either $\sigma'(u_0)(a) < 1$ or $\sigma'(u_0)(b) < 1$. If $\sigma'(u_0)(a) < 1$, then

$$\sigma'(u_0)(a) > (\sigma'(u_0)(a))^2 > \cdots > (\sigma'(u_0)(a))^n.$$

Furthermore, we see that

$$\mathbb{P}^{\sigma}(v_0)(\mathfrak{C}(a^n, t)) = 0.5 \cdot \mu^{*n}(t) \text{ and } \mathbb{P}^{\sigma'}(u_0)(\mathfrak{C}(a^n, t)) = (\sigma'(u_0)(a))^n \cdot \mu^{*n}(t)$$

for $n > 1$. Take some n such that $(\sigma'(u_0)(a))^n < 0.5$. In that case we get $\mathbb{P}^{\sigma'}(u_0)(\mathfrak{C}(a^n, t)) < \mathbb{P}^{\sigma}(v_0)(\mathfrak{C}(a^n, t))$. The same procedure can be used in case $\sigma'(u_0)(b) < 1$. Hence we conclude that $U \not\precsim V$, and therefore also that $U \not\equiv V$. ♦

Example C.4.11 also works for schedulers with memory, although the argument has to be modified a bit. In that case, in each step either the probability of a trace consisting only of a's or the probability of a trace consisting only of b's must decrease in U, so after some number of steps, the probability of one of these two must decrease below 0.5, and then the rest of the argument is the same.

Example C.4.12. Consider the SMDPs U and V in Figure C.4.1 and let $F_\mu = Exp[\theta_1]$ and $F_\nu = Exp[\theta_2]$ be exponential distributions with rates $\theta_1 > \theta_2 > 0$.

Then, as shown in Example C.4.9, it holds that $U \preceq V$. However, we have both $U \not\precsim V$ and $U \not\sim V$. ♦

From Examples C.4.11 and C.4.12, we get the following theorem.

Theorem C.4.13. *$\precsim$ and $\preceq$ are incomparable, $\sim$ and $\preceq$ are incomparable, and we have $\sim \not\subseteq \equiv$.*

C.5 Approximation

It has been shown in [19] that the faster-than relation is undecidable for the generative case. A small modification of the argument shows that the same is true for the reactive case.

Theorem C.5.1. *It is undecidable whether $U \preceq V$.*

Proof. The result follows from the fact that $U \preceq V$ is undecidable for the generative case. Let $U = (S_U, \tau, \rho)$ be a generative Markov process with set of actions L and construct the reactive Markov process $V = (S_V, \tau', \rho')$ as follows. Let $L' = L \cup \{\sharp\}$, where $\sharp$ is a new symbol not in L. For every state $s \in S_U$, we let $s^a \in S_V$ for every symbol $a \in L'$.

$$\tau'\left(s_1^a, a'\right)\left(s_2^{a''}\right) = \begin{cases} \tau(s_1)(a', s_2) & \text{if } a = a' \in L \text{ and } a'' = \sharp \\ \frac{1}{|L'|} & \text{if } a = a' = \sharp \text{ and } a'' \in L. \end{cases}$$

So each state s^a only has one outgoing action, namely a, and hence controllers play no role in the probabilities of V. Finally, let $\rho'(s^a) = \rho(s)$. Then we have

$$P_U(s)(\mathfrak{C}(a_1 \dots a_n, t)) = |L'|^n P_V\left(s^\sharp\right)(\mathfrak{C}(\sharp a_1 \sharp \dots \sharp a_n, t)),$$

and hence

$$P_U(s_1)(\mathfrak{C}(a_1 \dots a_n, t)) \geq P_U(s_2)(\mathfrak{C}(a_1 \dots a_n, t))$$

if and only if

$$P_V\left(s_1^\sharp\right)(\mathfrak{C}(\sharp a_1 \sharp \dots \sharp a_n, t)) \geq P_V\left(s_2^\sharp\right)(\mathfrak{C}(\sharp a_1 \sharp \dots \sharp a_n, t)).$$

This means that $s_1 \leq s_2$ if and only if $s_1^\sharp \leq s_2^\sharp$. ■

In view of Theorem C.5.1, we can extend the approximation algorithm for the generative case from [19] to the reactive case. In order to do this, we need to also consider the schedulers that are necessary for reactive systems. Instead of deciding the faster-than relation, we consider the time-bounded approximation problem, which asks the following: Given $\varepsilon > 0$, a time bound

$b \in \mathbb{R}_{\geq 0}$, and two SMDPs U and V, determine whether for all schedulers σ there exists a scheduler σ' such that

$$\mathbb{P}^{\sigma'}(u_0)(C) \geq \mathbb{P}^{\sigma}(v_0)(C) - \varepsilon \tag{C.1}$$

for all time-bounded cylinders $C = \mathfrak{C}(a_1 \dots a_n, t)$ where $t \leq b$.

First we identify the kind of distributions for which our algorithm will work. Given a class $\mathcal{C}$ of distributions, we let $\mathrm{Convex}(\mathcal{C})$ denote the closure of $\mathcal{C}$ under convex combinations, and $\mathrm{Conv}(\mathcal{C})$ denote the closure of $\mathcal{C}$ under both convex combinations and convolutions.

Definition C.5.2. A class of distributions $\mathcal{C}$ is *effective* if for any $\varepsilon > 0$, $b \in \mathbb{R}_{\geq 0}$, and $\mu_1, \mu_2 \in \mathrm{Conv}(\mathcal{C})$, $\{t \in \mathbb{R}_{\geq 0} \mid \mu_1([0,t]) \geq \mu_2([0,t]) - \varepsilon \text{ and } t \leq b\}$ is a semialgebraic set. ▲

A semialgebraic set is essentially one that can be expressed in the first-order theory of the reals, and hence membership in such a set can be decided by utilising the decidability of the first-order theory of reals [25]. In addition to effectiveness, we will also require residence-time distributions to take some non-zero amount of time to fire. This requirement is made precise by the following definition.

Definition C.5.3. A class $\mathcal{C}$ of distributions is *slow* if for any finite subset $\mathcal{C}_0 \subseteq \mathcal{C}$, there exists a computable function $\varepsilon : \mathbb{N} \times \mathbb{R}_{\geq 0} \to [0,1]$ such that for all $n \in \mathbb{N}$, $t \in \mathbb{R}_{\geq 0}$ and $\mu_1, \dots, \mu_n \in \mathrm{Convex}(\mathcal{C}_0)$ we have

$$(\mu_1 * \cdots * \mu_n)([0,t]) \leq \varepsilon(n,t)$$

and $\lim_{n\to\infty} \varepsilon(n,t) = 0$. ▲

It has been shown in [19] that the class of uniform distributions and the class of exponential distributions are both effective and slow. The importance of the closure under convex combinations and convolutions in Definition C.5.2 is explained by the following lemma.

Lemma C.5.4. *Let $\mathcal{C}$ be a class of distributions, and let U be a SMDP with residence-time distributions taken from $\mathcal{C}$. Then $\mathbb{P}^{\sigma}(s)(a_1 \dots a_n) \in \mathrm{Conv}(\mathcal{C})$ for any scheduler σ, state s, and $a_1, \dots, a_n \in L$.*

Proof. The lemma follows essentially from Proposition C.4.8. For $n = 1$ we get

$$\mathbb{P}^{\sigma}(s)(a) = \sum_{s' \in S} \tau^{\sigma}(s,a)(s') \cdot \rho(s) \in \mathrm{Conv}(\mathcal{C}).$$

For $n > 1$ we get

$$\mathbb{P}^{\sigma}(s)(a_1 \dots a_n) = \sum_{s' \in S} \tau^{\sigma}(s,a)(s') \cdot (\rho(s) * \mathbb{P}^{\sigma}(s')(a_2 \dots a_n)),$$

and since $\mathbb{P}^{\sigma}(s')(a_2 \dots a_n) \in \mathrm{Conv}(\mathcal{C})$ by induction hypothesis, it follows that $\mathbb{P}^{\sigma}(s)(a_1 \dots a_n) \in \mathrm{Conv}(\mathcal{C})$. ■

Lemma C.5.4 shows that, for fixed schedulers σ and σ', we can decide whether $\mathbb{P}^{\sigma}(u)(C) \geq \mathbb{P}^{\sigma'}(v)(C)$ whenever U and V have effective residence-time distributions using the first-order theory of reals.

The following theorem shows that, again for fixed schedulers, we can find an $N \in \mathbb{N}$ such that the probability of any time-bounded cylinder with length greater than N is less than ε. Therefore any such time-bounded cylinder trivially satisfies the inequality (C.1) and can thus be disregarded.

Theorem C.5.5 ([19, Theorem 5])**.** *Let U be a SMDP with slow residence-time distributions. For any state $s \in S$, $\varepsilon > 0$, $b \in \mathbb{R}_{\geq 0}$, and scheduler σ, there exists $N \in \mathbb{N}$ such that $\mathbb{P}^{\sigma}(s)(\mathfrak{C}(a_1 \dots a_n, b)) \leq \varepsilon$ for all $n \geq N$.*

All that is left now is to consider schedulers. However, since we only need to consider time-bounded cylinders up to some finite length, we can also represent a scheduler as a collection of finitely many probability distributions over the action labels. Each such distribution can in turn be represented as a collection of real variables that must sum to no more than 1. Hence schedulers can also be represented in the first-order theory of reals.

Theorem C.5.6. *Let U be a SMDP with slow residence-time distributions. Then the time-bounded approximation problem is decidable.*

Proof. By Theorem C.5.5, we can find some $N \in \mathbb{N}$ such that $\mathbb{P}^{\sigma'}(v_0)(C) - \varepsilon \leq 0$ for any scheduler σ' and any time-bounded cylinder bounded by b and of length $n \geq N$. This means that for any such time-bounded cylinder, we trivially have

$$\mathbb{P}^{\sigma}(u_0)(C) \geq \mathbb{P}^{\sigma'}(v_0)(C) - \varepsilon$$

for any scheduler σ. It is therefore enough to only consider time-bounded cylinders of length $n \leq N$.

Now let σ be a scheduler. We can represent σ in the first-order theory of reals as follows. For each state s and label a (recall there are finitely many of these), let $x_{s,a}$ be a real variable. Then we interpret $x_{s,a}$ to be the probability $\sigma(s)(a)$, and we impose the constraint $\sum_{a \in L} x_{s,a} \leq 1$. The whole time-bounded approximation problem can therefore be encoded in the first-order theory of reals, and is thus decidable. ■

C.6 Compositionality

Next we introduce the notion of composition of SMDPs. As argued in [23], the style of synchronous CSP composition is the most natural one to consider for reactive probabilistic systems, so this is the one we will adopt. However, we leave the composition of the residence-times as a parameter, so that we can compare different kinds of composition.

Definition C.6.1. A function $\star : \Delta(\mathbb{R}_{\geq 0}) \times \Delta(\mathbb{R}_{\geq 0}) \to \Delta(\mathbb{R}_{\geq 0})$ is called a *residence-time composition function* if it is commutative, i.e. $\star(\mu, \nu) = \star(\nu, \mu)$ for all $\mu, \nu \in \Delta(\mathbb{R}_{\geq 0})$. ▲

One example of such a composition function is when $\star$ is a coupling, which is a joint probability measure such that its marginals are μ and ν. A simple special case of this is the product measure $\star(\mu, \nu) = \mu \times \nu$, which is defined by $(\mu \times \nu)(B_1 \times B_2) = \mu(B_1) \cdot \nu(B_2)$ for all Borel B_1 and B_2.

In order to model the situation in which we want the composite system only to take a transition when both components can take a transition, it is natural to take the minimum of the two probabilities, which corresponds to waiting for the slowest of the two. In that case, we let

$$F_{\star(\mu,\nu)}(t) = \min\{F_\mu(t), F_\nu(t)\},$$

and we call this *minimum composition*. Likewise, if we only require one of the components to be able to take a transition, then it is natural to take the maximum of the two probabilities by letting

$$F_{\star(\mu,\nu)}(t) = \max\{F_\mu(t), F_\nu(t)\},$$

which we call *maximum composition*. A special case of minimum composition is the composition on rates used in PEPA [10], and a special case of maximum composition is the composition on rates used in TIPP [7].

Further knowledge about the processes that are being composed lets one define more specific composition functions. As an example, if we know that the components only have exponential distributions, then we can define composition functions that work directly on the rates of the distributions. If $F_\mu = Exp[\theta]$ and $F_\nu = Exp[\theta']$, then one could for example let $\star(\mu, \nu)$ be such that

$$F_{\star(\mu,\nu)} = Exp\left[\theta \cdot \theta'\right].$$

This corresponds to the composition on rates that is used in SPA [9], and we will call it *product composition*. Note that product composition is not given by the product measure.

Definition C.6.2. Let $\star$ be a residence-time composition function. Then the *$\star$-composition* of U and V, denoted by $U \parallel_\star V = (S, \tau, \rho)$, is given by

- $S = U \times V$,
- $\tau((u, v), a)((u', v')) = \tau_U(u, a)(u') \cdot \tau_V(v, a)(v')$ for all $a \in L$ and $(u', v') \in S$, and
- $\rho((u, v)) = \star(\rho_U(u), \rho_V(v))$. ▲

When considering the composite SMDP $U \parallel_\star V$ of two SMDPs U and V, we will also write $u \parallel_\star v$ to denote the composite state (u, v) of $U \parallel_\star V$ where $u \in S_U$ and $v \in S_V$.

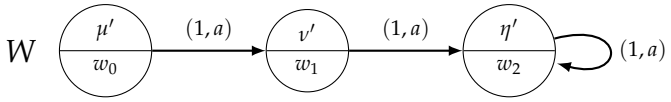


Figure C.6.1: For different instantiations of μ', ν', and η', the context W leads to parallel timing anomalies for product, minimum, and maximum rate composition, respectively.

C.6.1 Parallel Timing Anomalies

If we have two components U and V, and we know that U is faster than V, then if V is in parallel with some context W, we would expect this composition to become faster when we replace the component V with the component U. However, sometimes this fails to happen, and we will call such an occurrence a *parallel timing anomaly*.

In this section we show that parallel timing anomalies can occur for some of the kinds of composition discussed in Section C.6. We do this by giving different contexts W for the SMDPs U and V from Figure C.4.1, for which it was shown in Example C.4.9 that $U \preceq V$. Our examples of parallel timing anomalies make no use of non-determinism or probabilistic branching, thus showing that the parallel timing anomalies are caused inherently by the timing behaviour of the SMDPs. For ease of presentation, we let the set of labels L consist only of the label a in this section.

Consider the two SMDPs U and V depicted in Figure C.4.1. For the examples in this section, let $F_\mu = Exp[2]$, $F_\nu = Exp[0.5]$, and let η be arbitrary.

Example C.6.3 (Product composition)**.** Let $\star$ be product composition and let the context (W, w_0) be given by Figure C.6.1, where $F_{\mu'} = Exp[10]$, $F_{\nu'} = Exp[0.1]$ and $\eta = \eta'$. In $U \parallel_\star W$, the rates in the first two states will then be 20 and 0.05, and in $V \parallel_\star W$ they will be 5 and 0.5. Consider the time-bounded cylinder $\mathfrak{C}(aa, 2)$. Then we see that

$$\mathbb{P}(u_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.09 \quad \text{and} \quad \mathbb{P}(v_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.30,$$

showing that $U \parallel_\star W \not\preceq V \parallel_\star W$. Hence we have a parallel timing anomaly. What happens is that in the process $V \parallel_\star W$ the probability of taking a transition before time 2 with rate 5 is already very close to 1, so the process $U \parallel_\star W$ does not gain much by having a rate of 20, whereas in the next step, $V \parallel_\star W$ gains a lot of probability by having a rate of 0.5 compared to the rate 0.05 of $U \parallel_\star W$. ♦

Example C.6.4 (Minimum composition)**.** Let $\star$ be minimum composition and let the context (W, w_0) be given by Figure C.6.1, where $F_{\mu'} = Exp[1]$, $F_{\nu'} = Exp[2]$, and $\eta = \eta'$. The rates of $U \parallel_\star W$ are then 1 and 0.5, whereas they are 0.5 and 2 in $V \parallel_\star W$. Then

$$\mathbb{P}(u_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.40 \quad \text{and} \quad \mathbb{P}(v_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.51,$$

so $U \parallel_\star W \not\preceq V \parallel_\star W$. What happens in this example is that in the second step, $U \parallel_\star W$ has the same rate as $V \parallel_\star W$ had in the first step. This means that $U \parallel_\star W$ must be proportionally faster in the second step. However, $V \parallel_\star W$ has a rate of 2 in the second step, but $U \parallel_\star W$ only had a rate of 1 in the first step. ⧫

Example C.6.5 (Maximum composition)**.** Let $\star$ be maximum composition and let the context (W, w_0) be given by Figure C.6.1, where $F_{\mu'} = Exp[2]$, $F_{\nu'} = Exp[1]$, and $\eta = \eta'$. $U \parallel_\star W$ then has rates 2 and 1, and $V \parallel_\star W$ has rates 2 and 2. Then

$$\mathbb{P}(u_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.75 \quad \text{and} \quad \mathbb{P}(v_0 \parallel_\star w_0)(\mathfrak{C}(aa, 2)) \approx 0.91,$$

so $U \parallel_\star W_3 \not\preceq V \parallel_\star W_3$. The reason for the timing anomaly in this case is clear: $V \parallel_\star W$ simply has a higher rate in each step than $U \parallel_\star W$ does. ⧫

C.6.2 Avoiding Parallel Timing Anomalies

We have seen in the previous section that parallel timing anomalies can occur. We now wish to understand what kind of contexts do not lead to timing anomalies. In this section we assume that the set L of transition labels is a finite set. Also, we fix a residence-time composition function $\star$ and two additional SMDPs $(W, w_0) = (S_W, \tau_W, \rho_W)$ and $(W', w_0') = (S_{W'}, \tau_{W'}, \rho_{W'})$ which should be thought of as contexts. Next we identify conditions on (W, w_0) such that $U \preceq V$ will imply $U \parallel_\star W \preceq V \parallel_\star W$.

We first give conditions that over-approximate the faster-than relation between the composite systems by requiring that when U and W are put in parallel, then the composite system is point-wise faster than U along all paths. Likewise, we require that when V and W are put in parallel, the composite system is point-wise slower than V along all paths. If we already know that U is faster than V, this will imply by transitivity that $U \parallel_\star W$ is faster than $V \parallel_\star W$. We have already seen in Example C.4.9 that a process U need not be point-wise faster than V along all paths in order for U to be faster than V. However, by imposing this condition, we do not need to compare convolutions of distributions, but can compare the distributions directly.

We first introduce some terminology. We will say that a SMDP M has a *deterministic Markov kernel* if for all states s and labels a, there is at most one state s' such that $\tau(s, a)(s') > 0$.

Definition C.6.6. A *state path in* M is a sequence of states $s_1, s_2, \ldots$ where for all $i \in \mathbb{N}$ there exists a label $a \in L$ such that $\tau(s_i, a)(s_{i+1}) > 0$. For a state path $\pi = s_1, s_2, \ldots$, we let $\pi[i] = s_i$, $\pi|^i = s_i, s_{i+1}, \ldots$, $\pi|_i = s_1, s_2, \ldots, s_i$, and we let $\Pi[M]$ denote the set of all state paths in M. For a state $s \in S$, we let $\Pi[s] = \{\pi \in \Pi[M] \mid \pi[1] = s\}$ and we let $\Pi_n[s] = \{\pi|_n \mid \pi \in \Pi[s]\}$. ▲

Definition C.6.7. Let $n \in \mathbb{N}$. We say that $\star$ is *n-monotonic in U, V, W, and W'*, written $(U, W) \lessapprox_\star^n (V, W')$, if W' has a deterministic Markov kernel and

- $F_{\rho(\pi_U[i] \|_\star \pi_W[i])}(t) \geq F_{\rho_U(\pi_U[i])}(t)$ and $F_{\rho_V(\pi_V[i])}(t) \geq F_{\rho(\pi_V[i] \|_\star \pi_{W'}[i])}(t)$ for all $t \in \mathbb{R}_{\geq 0}$ and $1 \leq i \leq n$,
- for all schedulers σ_U for U there exists a scheduler $\sigma_{U,W}$ for $U \|_\star W$ such that we have

$$\tau^{\sigma_{U,W}}(\pi_U[i] \|_\star \pi_W[i], a)(\pi_U[i+1] \|_\star \pi_W[i+1]) \geq \tau_U^{\sigma_U}(\pi_U[i], a)(\pi_U[i+1]),$$

  and

- for all schedulers $\sigma_{V,W'}$ for $V \|_\star W'$ there exists a scheduler σ_V for V such that we have

$$\tau_V^{\sigma_V}(\pi_V[i], a)(\pi_V[i+1]) \geq \tau^{\sigma_{V,W'}}(\pi_V[i] \|_\star \pi_{W'}[i], a)(\pi_V[i+1] \|_\star \pi_{W'}[i+1])$$

for all state paths $\pi_U \in \Pi_n[u_0]$, $\pi_V \in \Pi_n[v_0]$, $\pi_W \in \Pi_n[w_0]$, and $\pi_{W'} \in \Pi_n[w_0']$, and for all $a \in L$ and $1 \leq i < n$. Furthermore, we will say that $\star$ is *monotonic in U, V, W, and W'* and write $(U, W) \lessapprox_\star (V, W')$, if it is n-monotonic in U, V, W, and W' for all $n \in \mathbb{N}$. ▲

Clearly, if $(U, W) \lessapprox_\star^n (V, W')$, then $(U, W) \lessapprox_\star^m (V, W')$ for all $m \leq n$. The next result shows that if $(U, W) \lessapprox_\star (V, W')$, then we are guaranteed to avoid parallel timing anomalies.

Theorem C.6.8. *If $(U, W) \lessapprox_\star (V, W')$ as well as $U \preceq V$ and $W \preceq W'$, then we have $U \|_\star W \preceq V \|_\star W'$.*

Proof. Let $\mathfrak{C}(a_1 \dots a_n, x)$ be an arbitrary time-bounded cylinder, and let $\sigma_{V,W'}$ be an arbitrary scheduler for $V \|_\star W'$. Because $(U, W) \lessapprox_\star (V, W')$, there exists a scheduler σ_V for V and a path π such that

$$\begin{aligned}
&\mathbb{P}^{\sigma_{V,W'}}(v_0 \|_\star w_0')(\mathfrak{C}(a_1 \dots a_n, t)) \\
&= \tau^{\sigma_{V,W'}}(\pi[1], a_1)(\pi[2]) \cdots \tau^{\sigma_{V,W'}}(\pi[n], a_n)(\pi[n+1]) \\
&\quad \cdot (\rho(\pi[1]) * \cdots * \rho(\pi[n]))([0, t]) \\
&\leq \tau_V^{\sigma_V}(\pi_V[1], a_1)(\pi_V[2]) \cdots \tau_V^{\sigma_V}(\pi_V[n], a_n)(\pi[n+1]) \\
&\quad \cdot (\rho_V(\pi_V[1]) * \cdots * \rho_V(\pi_V[n]))([0, t]) \\
&\leq \sum_{\pi \in \Pi_{n+1}[v_0]} \tau_V^{\sigma_V}(\pi[1], a_1)(\pi[2]) \cdots \tau_V^{\sigma_V}(\pi[n], a_n)(\pi[n+1]) \\
&\quad \cdot (\rho_V(\pi[1]) * \cdots * \rho_V(\pi[n]))([0, t]) \\
&= \mathbb{P}^{\sigma_V}(v_0)(\mathfrak{C}(a_1 \dots a_n, t)),
\end{aligned}$$

Since $U \preceq V$, there must exist some scheduler σ_U for U such that

$$\mathbb{P}^{\sigma_V}(v_0)(\mathfrak{C}(a_1 \dots a_n, t))) \leq \mathbb{P}^{\sigma_U}(u_0)(\mathfrak{C}(a_1 \dots a_n, t)).$$

Again, since $(U, W) \lessapprox_\star (V, W')$, there exists a scheduler $\sigma_{U,W}$ for $U \parallel_\star W$ such that

$$\begin{aligned}
&\mathbb{P}^{\sigma_U}(u_0)(\mathfrak{C}(a_1 \dots a_n, t)) \\
&= \sum_{\pi \in \Pi_{n+1}[u_0]} \tau_U^{\sigma_U}(\pi[1], a_1)(\pi[2]) \cdots \tau_U^{\sigma_U}(\pi[n], a_n)(\pi[n+1]) \\
&\quad \cdot (\rho_U(\pi[1]) * \cdots * \rho_U(\pi[n]))([0, t]) \\
&\leq \sum_{\pi_W \in \Pi_{n+1}[w_0]} \sum_{\pi \in \Pi_{n+1}[u_0]} \tau_U^{\sigma_U}(\pi[1], a_1)(\pi[2]) \cdots \tau_U^{\sigma_U}(\pi[n], a_n)(\pi[n+1]) \\
&\quad \cdot (\rho_U(\pi[1]) * \cdots * \rho_U(\pi[n]))([0, t]) \\
&\leq \sum_{\pi \in \Pi_{n+1}[u_0 \parallel_\star w_0]} \tau^{\sigma_{U,W}}(\pi[1], a_1)([2]) \cdots \tau^{\sigma_{U,W}}([n], a_n)([n+1]) \\
&\quad \cdot (\rho(\pi[1]) * \cdots * \rho(\pi[n]))([0, t]) \\
&= \mathbb{P}^{\sigma_{U,W}}(u_0 \parallel_\star w_0)(\mathfrak{C}(a_1 \dots a_n, t)).
\end{aligned}$$

■

The special case where $W = W'$ shows that this condition is sufficient to avoid parallel timing anomalies. We do not know if it is decidable whether $(U, W) \lessapprox_\star (V, W')$. However, there is a stronger condition which is decidable in the case of finite SMDPs. We present it in the next definition.

Definition C.6.9. We say that $\star$ is *strongly* n-monotonic in U, V, W, and W' and write $(U, W) \leqq^n_\star (V, W')$ if W' has a deterministic Markov kernel and for all state paths $\pi_U \in \Pi_n[u_0]$, $\pi_V \in \Pi_n[v_0]$, $\pi_W \in \Pi_n[w_0]$, and $\pi_{W'} \in \Pi_n[w_0']$, the first condition of Definition C.6.7 is satisfied and

- for all schedulers σ_U for U and all schedulers $\sigma_{U,W}$ for $U \parallel_\star W$, it is the case that

$$\tau^{\sigma_{U,W}}(\pi_U[i] \parallel_\star \pi_W[i], a)(\pi_U[i+1] \parallel_\star \pi_W[i+1]) \geq \tau_U^{\sigma_U}(\pi_U[i], a)(\pi_U[i+1]),$$

  and

- for all schedulers $\sigma_{V,W'}$ for $V \parallel_\star W'$ and all schedulers σ_V for V, it is the case that

$$\tau_V^{\sigma_V}(\pi_V[i], a)(\pi_V[i+1]) \geq \tau^{\sigma_{V,W'}}(\pi_V[i] \parallel_\star \pi_{W'}[i], a)(\pi_V[i+1] \parallel_\star \pi_{W'}[i+1])$$

for all $a \in L$ and $1 \leq i < n$. If $(U, W) \leqq^n_\star (V, W')$ for all $n \in \mathbb{N}$, we say that $\star$ is *strongly* monotonic in U, V, W, and W' and write $(U, W) \leqq_\star (V, W')$. ▲

The conditions of Definition C.6.9 are the second and third conditions from Definition C.6.7 with the existential quantifier strengthened to a universal quantifier. It is obvious that $(U, W) \leqq_\star (V, W')$ implies $(U, W) \lessapprox_\star (V, W')$, and hence we get the following corollary.

Corollary C.6.10. *If $(U, W) \leqq_\star (V, W')$ as well as $U \preceq V$ and $W \preceq W'$, then $U \parallel_\star W \preceq V \parallel_\star W'$.*

Example C.6.11. Let U and V be given by Figure C.4.1 with $F_\mu \geq F_\nu$ as in Example C.4.9. Let $\star$ be minimum rate composition and consider the context W from Figure C.6.1, where $\mu' = \mu$, $\nu' = \nu$, and $\eta' = \eta$. There is only one possible scheduler σ, which is the Dirac measure at a, and hence it is clear that the second and third conditions are satisfied. We also find that

$$\begin{aligned} F_{\rho(u_0 \parallel_\star w_0)}(t) &= F_{\rho_U(u_0)}(t) & F_{\rho_V(v_0)}(t) &= F_{\rho(v_0 \parallel_\star w_0)}(t) \\ F_{\rho(u_1 \parallel_\star w_1)}(t) &= F_{\rho_U(u_1)}(t) & F_{\rho_V(v_1)}(t) &= F_{\rho(v_1 \parallel_\star w_1)}(t) \\ F_{\rho(u_2 \parallel_\star w_2)}(t) &= F_{\rho_U(u_2)}(t) & F_{\rho_V(v_2)}(t) &= F_{\rho(v_2 \parallel_\star w_2)}(t) \end{aligned}$$

and hence the first condition is also satisfied, so $(U, W) \leqq_\star (V, W)$. ⧫

Example C.6.12. All the examples we gave in Section C.6.1 are not monotonic, and hence also not strongly monotonic, since they all violate condition 1 of monotonicity.

In Example C.6.3, this is because

$$F_{\rho(v_0)}(t) = Exp[0.5]\,(t) < Exp[5]\,(t) = F_{\rho(v_0 \parallel_\star w_0)}(t)$$

for any $t > 0$. Likewise, in Example C.6.4 we have

$$F_{\rho(u_0 \parallel_\star w_0)}(t) = Exp[1]\,(t) < Exp[2]\,(t) = F_{\rho(u_0)}(t)$$

for any $t > 0$. Finally, in Example C.6.5 we have

$$F_{\rho(v_0)}(t) = Exp[0.5]\,(t) < Exp[2]\,(t) = F_{\rho(v_0 \parallel_\star w_0)}(t)$$

for any $t > 0$. ⧫

We now wish to show that it is decidable whether $(U, W) \leqq_\star (V, W')$ for finite SMDPs, thereby giving a decidable condition for avoiding timing anomalies. To do this, we first show that in order to establish strong monotonicity, it is enough to consider paths up to length

$$m = \max\{|S_U| \cdot |S_W|, |S_V| \cdot |S_{W'}|\} + \max\{|S_U|, |S_V|, |S_W|, |S_{W'}|\} + 1,$$

due to the fact that they start repeating.

Lemma C.6.13. *Let U and V be two finite, pointed SMDPs. For any state paths π_U and π_V of length $l > |S_U| \cdot |S_V|$, there will be $i < j \leq |S_U| \cdot |S_V| + 1$ such that $\pi_U[i] = \pi_U[j]$, $\pi_V[i] = \pi_V[j]$.*

Proof. Since there are $|S_U| \cdot |S_V|$ ways of choosing a pair $(u_i, v_j) \in S_U \times S_V$ of states from U and V, if we pair the states of π_U and π_V such that we get the pairs $(\pi_U[1], \pi_V[1]), (\pi_U[2], \pi_V[2]), \ldots, (\pi_U[l], \pi_V[l])$, there must be two of these pairs that are the same because $l > |S_U| \cdot |S_V|$. Hence we get states $\pi_U[i] = \pi_U[j]$ and $\pi_V[i] = \pi_V[j]$ with $i < j \leq n$. It also follows that i and j can be chosen so that $i < j \leq |S_U| \cdot |S_V|$, because otherwise we would have $j - i > |S_U| \cdot |S_V|$ different pairs

$$(\pi_U[i], \pi_V[i]), (\pi_U[i+1], \pi_V[i+1]), \ldots, (\pi_U[j], \pi_V[j]),$$

contradicting the fact that there are only $|S_U| \cdot |S_V|$ such different pairs. ■

Lemma C.6.14. *Let (U, u_0), (V, v_0), (W, w_0), and (W', w_0') be finite, pointed SMDPs. If $(U, W) \leq^m_\star (V, W')$, then $(U, W) \leq_\star (V, W')$.*

Proof. Assume that $(U, W) \leq^m_\star (V, W')$. Then $(U, W) \leq^k_\star (V, W')$ for all $k \leq m$. Hence it remains to show that $(U, W) \leq^k_\star (V, W')$ for all $k > m$.

Let $k > m$ and consider two state paths $\pi_U = u_1 u_2 \ldots u_k$ and $\pi_W = w_1 w_2 \ldots w_k$ of U and W, respectively, both of length k. By Lemma C.6.13 there must exist $i < j \leq |S_U| \cdot |S_W| + 1$ such that $u_i = u_j$ and $w_i = w_j$. Since there exists a state path from u_1 to u_i, it must be possible to reach this state in less than $|S_U|$ steps, and likewise for W. Hence there must exist $l \leq \max\{|S_U|, |S_W|\}$, and state paths

$$u_1 u_2' \ldots u_l' u_i u_{i+1} \ldots u_j \quad \text{and} \quad w_1 w_2' \ldots w_l' w_i w_{i+1} \ldots w_j$$

of length $l + (j - i) \leq \max\{|S_U|, |S_W|\} + |S_U| \cdot |S_W| + 1 \leq m$. Hence we know that the conditions of Definition C.6.9 are satisfied for $u_i \ldots u_j$ and $w_i \ldots w_j$. By removing the states $u_{i+1} \ldots u_j$ and $w_{i+1} \ldots w_j$ from π_U and π_W we end up with two new state paths π_U' and π_W' of length $k' = k - (j - i)$. We can keep doing this as long as $k' > m$, so at some point we must end up with state paths π_U^* and π_W^* of length $k^* \leq m$, for which the conditions of Definition C.6.9 are satisfied by assumption, and hence they are satisfied for all of π_U and π_W. The same argument can be applied to two state paths π_V and $\pi_{W'}$ of V and W', so we conclude that $(U, W) \leq_\star (V, W')$. ■

We can now use the first-order theory of the reals to show that strong monotonicity is a decidable property.

Theorem C.6.15. *Consider the finite pointed SMDPs (U, u_0), (V, v_0), (W, w_0), and (W', w_0'). If for all state paths $\pi_U \in \Pi_m[u_0]$, $\pi_V \in \Pi_m[v_0]$, $\pi_W \in \Pi_m[w_0]$, and $\pi_{W'} \in \Pi_m[w_0']$ we have that $\{t \in \mathbb{R}_{\geq 0} \mid F_{\rho(\pi_U[i] \parallel_\star \pi_W[i])}(t) \geq F_{\rho_U(\pi_U[i])}(t)\}$ and $\{t \in \mathbb{R}_{\geq 0} \mid F_{\rho_V(\pi_V[i])}(t) \geq F_{\rho(\pi_V[i] \parallel_\star \pi_{W'}[i])}(t)\}$ are semialgebraic sets for all $1 \leq i \leq m$, then it is decidable whether $(U, W) \leq_\star (V, W')$.*

Proof. Note first of all that since L and W' are finite, it is decidable whether W' has a deterministic Markov kernel by looking at all the states. By Lemma C.6.14, it suffices to check whether $(U, W) \leqq^{m}_{\star} (V, W')$ where

$$m = \max\{|S_U| \cdot |S_W|, |S_V| \cdot |S_{W'}|\} + \max\{|S_U|, |S_V|, |S_W|, |S_{W'}|\} + 1.$$

This can be done by exploiting the decidability of the first-order theory of the reals in the following way. Since L is finite and U, V, W, and W' are all finite, there are finitely many state paths $\pi_U \in \Pi_m[u_0]$, $\pi_V \in \Pi_m[v_0]$, $\pi_W \in \Pi_m[w_0]$, and $\pi_{W'} \in \Pi_m[w'_0]$. Because of this, and since the sets

$$\{t \in \mathbb{R}_{\geq 0} \mid F_{\rho(\pi_U[i] \|_\star \pi_W[i])}(t) \geq F_{\rho_U(\pi_U[i])}(t)\}$$

and

$$\{t \in \mathbb{R}_{\geq 0} \mid F_{\rho_V(\pi_V[i])}(t) \geq F_{\rho(\pi_V[i] \|_\star \pi_{W'}[i])}(t)\},$$

which we need to check for the first condition, were assumed to be semi-algebraic, it is possible to express the conditions of Definition C.6.9 in the first-order theory of the reals, using finitely many quantifiers and inequalities. Since the first-order theory of the reals is decidable, the truth value of the resulting formula is decidable. ∎

For uniform and exponential distributions with minimum or maximum composition, the corresponding sets are all semialgebraic, and the same is true for exponential distributions with product composition. Theorem C.6.15 can therefore be used for these types of composition.

Unfortunately, strong monotonicity is a very strict requirement. In effect, it requires that there is only one possible action, and hence rules out non-determinism. However, strong monotonicity still makes sense as a requirement on processes with no non-determinism, since all our examples of timing anomalies in Section C.6.1 are of this form.

Proposition C.6.16. *If $(U, W) \leqq_\star (V, W')$, then L is a singleton set or u_0 is a deadlock state.*

Proof of Proposition C.6.16. We prove the contrapositive. Suppose $|L| > 1$ and u_0 is not a deadlock state. Because u_0 is not a deadlock state, there must exist some state path π_U such that $\pi_U[1] = u_0$ and $\tau_U(\pi_U[1], a)(\pi_U[2]) > 0$ for some $a \in L$. Since $|L| > 1$, we can find some $b \in L$ with $a \neq b$. Now construct schedulers given by $\sigma_{U,W}(s) = \delta_b$ and $\sigma_U(s) = \delta_a$ for any state s. Then

$$\tau^{\sigma_{U,W}}(\pi_U[1] \parallel_\star \pi_W[1], a)(\tau_U[2] \parallel_\star \tau_W[2]) = 0$$

but

$$\tau_U^{\sigma_U}(\pi_U[1], a)(\tau_U[2]) > 0,$$

and hence the first condition of Definition C.6.9 is violated. ∎

C.7 Conclusion

In this paper, we have investigated the notion of a process being faster than another process in the context of semi-Markov decision processes. We have given a trace-based definition of a faster-than relation, and shown that this definition is closely connected to convolutions of distributions. The faster-than relation is unfortunately undecidable, but we have shown how to approximate a time-bounded version of it. By considering composition as being parametric in how the residence times of states are combined, we have given examples showing that our faster-than relation gives rise to parallel timing anomalies for many of the popular ways of composing rates. We have therefore given sufficient conditions for how such parallel timing anomalies can be avoided, and we have shown that these conditions are decidable.

The main challenge that we face when trying to construct algorithms for the faster-than relation is that of schedulers, and in particular the juxtaposition between the universal and existential quantification over schedulers. For example, we had to strengthen the existential quantifier to a universal one in order to decide the conditions for avoiding parallel timing anomalies. This is because we know that locally, for any scheduler σ, there exists a scheduler σ' which works. However, it is not clear that all of these σ' can be collected coherently into a single scheduler which works globally. Solving this challenge would allow us to decide the property of monotonicity instead of the too-strong property of strong monotonicity, as well as prove decidability for so-called unambiguous processes.

The conditions we have given for avoiding timing anomalies do not look at the context in isolation, but depend also on the processes that are being swapped. It would be preferable to have conditions on a context that would guarantee the absence of parallel timing anomalies no matter what processes are being swapped.

C.8 References

[1] C. Baier, J. Katoen, H. Hermanns, and V. Wolf, "Comparative branching-time semantics for Markov chains," *Inf. Comput.*, vol. 200, no. 2, pp. 149–214, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2005.03.001

[2] P. Billingsley, *Probability And Measure*, 3rd ed. Wiley-Interscience, 1995.

[3] F. Cassez, R. R. Hansen, and M. C. Olesen, "What is a timing anomaly?" in *12th International Workshop on Worst-Case Execution Time Analysis, WCET 2012, July 10, 2012, Pisa, Italy*, ser. OASICS, T. Vardanega, Ed., vol. 23. Schloss Dagstuhl -

Leibniz-Zentrum fuer Informatik, 2012, pp. 1–12. [Online]. Available: https://doi.org/10.4230/OASIcs.WCET.2012.1

[4] E. M. Clarke, D. E. Long, and K. L. McMillan, "Compositional model checking," in *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 1989, pp. 353–362. [Online]. Available: https://doi.org/10.1109/LICS.1989.39190

[5] F. Corradini, R. Gorrieri, and M. Roccetti, "Performance preorder: Ordering processes with respect to speed," in *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, ser. Lecture Notes in Computer Science, J. Wiedermann and P. Hájek, Eds., vol. 969. Springer, 1995, pp. 444–453. [Online]. Available: https://doi.org/10.1007/3-540-60246-1_150

[6] M. Geilen, S. Tripakis, and M. Wiggers, "The earlier the better: a theory of timed actor interfaces," in *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, M. Caccamo, E. Frazzoli, and R. Grosu, Eds. ACM, 2011, pp. 23–32.

[7] N. Götz, U. Herzog, and M. Rettelbach, "Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras," in *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93, Santa Clara, CA, USA, May 10-14, 1993*, ser. Lecture Notes in Computer Science, L. Donatiello and R. D. Nelson, Eds., vol. 729. Springer, 1993, pp. 121–146.

[8] R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal of Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969. [Online]. Available: http://www.jstor.org/stable/2099572

[9] H. Hermanns, U. Herzog, and V. Mertsiotakis, "Stochastic process algebras - between LOTOS and Markov chains," *Computer Networks*, vol. 30, no. 9-10, pp. 901–924, 1998.

[10] J. Hillston, *A compositional approach to performance modelling*, ser. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996. [Online]. Available: https://doi.org/10.1017/CBO9780511569951

[11] R. Kirner, A. Kadlec, and P. P. Puschner, "Precise worst-case execution time analysis for processors with timing anomalies," in *21st Euromicro*

Conference on Real-Time Systems, ECRTS 2009, Dublin, Ireland, July 1-3, 2009. IEEE Computer Society, 2009, pp. 119–128. [Online]. Available: https://doi.org/10.1109/ECRTS.2009.8

[12] E. A. Lee, "Cyber physical systems: Design challenges," in *11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008), 5-7 May 2008, Orlando, Florida, USA*. IEEE Computer Society, 2008, pp. 363–369.

[13] T. Lundqvist and P. Stenström, "Timing anomalies in dynamically scheduled microprocessors," in *Proceedings of the 20th IEEE Real-Time Systems Symposium, Phoenix, AZ, USA, December 1-3, 1999*. IEEE Computer Society, 1999, pp. 12–21. [Online]. Available: https://doi.org/10.1109/REAL.1999.818824

[14] G. Lüttgen and W. Vogler, "A faster-than relation for asynchronous processes," in *CONCUR 2001 - Concurrency Theory, 12th International Conference, Aalborg, Denmark, August 20-25, 2001, Proceedings*, ser. Lecture Notes in Computer Science, K. G. Larsen and M. Nielsen, Eds., vol. 2154. Springer, 2001, pp. 262–276. [Online]. Available: https://doi.org/10.1007/3-540-44685-0_18

[15] ——, "Bisimulation on speed: A unified approach," *Theor. Comput. Sci.*, vol. 360, no. 1-3, pp. 209–227, 2006. [Online]. Available: https://doi.org/10.1016/j.tcs.2006.03.004

[16] S. Maovi, S. Stoi, and R. Hajdin, "Application of semi-Markov decision process in bridge management," *IABSE Symposium Report*, vol. 105, no. 28, pp. 1–8, 2015. [Online]. Available: https://www.ingentaconnect.com/content/iabse/report/2015/00000105/00000028/art00005

[17] F. Moller and C. M. N. Tofts, "Relating processes with respect to speed," in *CONCUR '91, 2nd International Conference on Concurrency Theory, Amsterdam, The Netherlands, August 26-29, 1991, Proceedings*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. F. Groote, Eds., vol. 527. Springer, 1991, pp. 424–438. [Online]. Available: https://doi.org/10.1007/3-540-54430-5_104

[18] M. R. Neuhäußer and J. Katoen, "Bisimulation and logical preservation for continuous-time Markov decision processes," in *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires and V. T. Vasconcelos, Eds., vol. 4703. Springer, 2007, pp. 412–427. [Online]. Available: https://doi.org/10.1007/978-3-540-74407-8

[19] M. R. Pedersen, N. Fijalkow, G. Bacci, K. G. Larsen, and R. Mardare, "Timed comparisons of semi-Markov processes," in *Language and Automata Theory and Applications - 12th International Conference, LATA 2018, Ramat Gan, Israel, April 9-11, 2018, Proceedings*, ser. Lecture Notes in Computer Science, S. T. Klein, C. Martín-Vide, and D. Shapira, Eds., vol. 10792. Springer, 2018, pp. 271–283. [Online]. Available: https://doi.org/10.1007/978-3-319-77313-1_21

[20] M. R. Pulungan, "Reduction of acyclic phase-type representations," Ph.D. dissertation, Faculty of Natural Sciences and Technology, Saarland University, 2009.

[21] J. Reineke, B. Wachter, S. Thesing, R. Wilhelm, I. Polian, J. Eisinger, and B. Becker, "A definition and classification of timing anomalies," in *6th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis, July 4, 2006, Dresden, Germany*, ser. OASICS, F. Mueller, Ed., vol. 4. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2006/671

[22] I. Satoh and M. Tokoro, "A formalism for remotely interacting processes," in *Theory and Practice of Parallel Programming, International Workshop TPPP'94, Sendai, Japan, November 7-9, 1994, Proceedings*, ser. Lecture Notes in Computer Science, T. Ito and A. Yonezawa, Eds., vol. 907. Springer, 1994, pp. 216–228. [Online]. Available: https://doi.org/10.1007/BFb0026571

[23] A. Sokolova and E. P. de Vink, "Probabilistic automata: System types, parallel composition and comparison," in *Validation of Stochastic Systems - A Guide to Current Research*, ser. Lecture Notes in Computer Science, C. Baier, B. R. Haverkort, H. Hermanns, J. Katoen, and M. Siegle, Eds., vol. 2925. Springer, 2004, pp. 1–43. [Online]. Available: https://doi.org/10.1007/978-3-540-24611-4_1

[24] T. Tao, *An Introduction to Measure Theory*, ser. Graduate studies in mathematics. American Mathematical Society, 2013.

[25] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*. University of California press, 1951.

[26] O. S. Thomas and J. O. Sobanjo, "Semi-Markov decision process: A decision tool for transportation infrastructure management systems," in *International Conference on Transportation and Development*, 2016. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/9780784479926.036

[27] W. Vogler, "Timed testing of concurrent systems," *Inf. Comput.*, vol. 121, no. 2, pp. 149–171, 1995. [Online]. Available: https://doi.org/10.1006/inco.1995.1130

[28] ——, "Faster asynchronous systems," *Inf. Comput.*, vol. 184, no. 2, pp. 311–342, 2003. [Online]. Available: https://doi.org/10.1016/S0890-5401(03)00065-8

Paper D

A Hemimetric Extension of Simulation for Semi-Markov Processes

Mathias R. Pedersen, Giorgio Bacci, Kim G. Larsen, and Radu Mardare

The paper has been published in the
Proceedings of the 15th International Conference on Quantitative Evaluation of Systems, pp. 339–355, 2018.

The layout has been revised and the content extended.

Abstract

Semi-Markov decision processes (SMDPs) are continuous-time Markov decision processes where the residence-time on states is governed by generic distributions on the positive real line.

In this paper we consider the problem of comparing two SMDPs with respect to their time-dependent behaviour. We propose a hemimetric between processes, which we call simulation distance*, measuring the least acceleration factor by which a process needs to speed up its actions in order to behave at least as fast as another process. We show that this distance can be computed in time* $\mathcal{O}(n^2(f(l)+k)+m^2n)$*, where* n *is the number of states,* m *the number of actions,* k *the number of atomic propositions, and* $f(l)$ *the complexity of comparing the residence-time between states. The theoretical relevance and applicability of this distance is further argued by showing that (i) it is suitable for compositional reasoning with respect to CSP-like parallel composition and (ii) has a logical characterisation in terms of a simple Markovian logic.*

D.1 Introduction

Semi-Markov decision processes (SMDPs) are Markovian stochastic decision processes modelling the firing time of transitions via real-valued random variables describing the residence-time on states. Semi-Markov decision processes provide a more permissive model than continuous-time Markov decision processes, since they allow as residence-time distributions any generic distribution on the positive real line, rather than only exponential ones. The generality offered by SMDPs has been found useful in modelling several real-case scenarios. Successful examples include power plants [20] and power supply units [21], to name a few.

When considering such real-time stochastic processes, non-functional requirements are important, particularly requirements like response time and throughput, which depend on the timing behaviour of the process. We therefore wish to understand and be able to compare the timing behaviour of different processes.

To cope with the need for comparing the timing behaviour of different systems, in this paper we propose and study a quantitative extension of the simulation relation by Baier et al. [2], called ε-simulation, which puts the focus on the timing aspect of processes. The intuition is that a process s_2 *ε-simulates* another process s_1 if after accelerating the actions of s_2 by a factor $\varepsilon > 0$ it reacts to the inputs from the external environment as s_1 with at least the same speed.

This type of quantitative reasoning is not new in the literature, and it dates back to the seminal work of Jou and Smolka [10, 16], who proposed the con-

cept of probabilistic ε-bisimulation. This line of work has lead to much work on probabilistic bisimulation distances [5, 8, 9]. While our work is conceptually similar to the bisimulation distances, it is technically very different. This is because bisimulation distances are constructed from a coalgebraic view as fixed points of operators. However, for the kind of timed systems that we are investigating, the coalgebraic perspective is much less understood. Moreover, since our distance generalises a preorder relation and not a congruence as the other distances do, it is not symmetric, which brings in new technical challenges

Following the work of Jou and Smolka, our notion of ε-simulation naturally induces a distance between processes: For any pair of states s_1 and s_2, we define their *simulation distance* as the least acceleration factor needed by s_2 to speed up its actions in order to behave at least as fast as s_1. This definition does not provide a distance in the usual sense, but rather a *multiplicative hemimetric*, i.e. an asymmetric notion of distance satisfying a multiplicative version of the triangle inequality. Such a notion is not new, as it is extensively applied in the context of differential privacy to measure information leakage of systems (see e.g. [1, 4]).

The theoretical relevance and applicability of the simulation distance is argued by means of the following results, which are the main technical contributions of the paper:

1. We provide an algorithm for computing the simulation distance between arbitrary states of an SMDP running in time $\mathcal{O}(n^2(f(l)+k)+m^2n)$, where n is the number of states, m the number of actions, k the number of atomic propositions, and $f(l)$ the complexity of comparing the residence time distributions on states.

2. We show that under some mild conditions on how residence-time distributions are combined in the parallel composition of two states, CSP-like parallel composition of SMDPs is non-expansive with respect to our hemimetric. This shows that the simulation distance is suitable for compositional reasoning.

3. We provide a logical characterisation of the distance in terms of a simple Markovian logic, stating that the distance from s_1 to s_2 is less than or equal to ε if and only if s_2 satisfies the ε-perturbation of any logical property that s_1 satisfies. Moreover, we prove that ε-simulation preserves the ε-perturbation of time-bounded reachability properties.

4. We show that sets of formulas in our logic are closed in the topology induced by the distance. This means that approximate reasoning is sound in the limit: If a sequence of state converging to a limit all satisfy some logical property, then the limit will also satisfy this property.

Notation and Preliminaries.

Let $\mathbb{N}$ denote the natural numbers, $\mathbb{Q}_{\geq 0}$ the non-negative rational numbers, $\mathbb{R}_{\geq 0}$ the non-negative real numbers, and $\mathbb{R}_{>0}$ the strictly positive ones. We equip the real numbers with the usual Borel σ-algebra. Given a set X, we will denote by $\mathcal{D}(X)$ the set of all probability measures on X. A probability measure $\mu \in \mathcal{D}(X)$ is said to be *finitely supported* if the set $\{x \in X \mid \mu(x) > 0\}$ is finite. If $\mu \in \mathcal{D}(\mathbb{R}_{\geq 0})$, then the *cumulative distribution function* (CDF) will be denoted by F_μ and is given by $F_\mu(t) = \mu([0,t])$. Any CDF F is increasing, meaning that if $t \geq t'$, then $F(t) \geq F(t')$, and also right-continuous, meaning that for all $\varepsilon > 0$ there exists a $\delta > 0$ such that if $t < t' < t + \delta$, then $|F(t') - F(t)| < \varepsilon$. For $x \in \mathbb{R}_{\geq 0}$, we will write δ_x for the Dirac measure at x, which is defined by $\delta_x(E) = 1$ if $x \in E$ and $\delta_x(E) = 0$ otherwise. For any $\theta \in \mathbb{R}_{>0}$, we will write $Exp[\theta]$ for the CDF of an exponential distribution with rate θ, and for $a, b \in \mathbb{R}_{\geq 0}$ such that $a < b$, we will write $Unif[a,b]$ for the CDF of a uniform distribution.

We will use the convention that $\infty + x = \infty$ for $x \in \mathbb{R}_{\geq 0}$ and $\infty \cdot y = \infty$ for $y \in \mathbb{R}_{>0}$. A function $d \colon X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is called a *hemimetric* if it satisfies $d(x,x) = 0$ and the triangle inequality $d(x,z) \leq d(x,y) + d(y,z)$. It is called a *pseudometric* if it is also symmetric, i.e. $d(x,y) = d(y,x)$, and it is called a *metric* if it is symmetric and furthermore $d(x,y) = 0$ if and only if $x = y$.

D.2 Semi-Markov Decision Processes

In this section, we introduce semi-Markov decision processes, which are continuous-time reactive probabilistic systems. A semi-Markov decision process has residence time on states governed by generic distributions on the positive real line and reacts to inputs from an external environment by making a probabilistic transition to a next state.

Hereafter, we consider a non-empty finite set of *input actions* A, and a non-empty, finite set of *atomic propositions* $\mathcal{AP}$.

Definition D.2.1. A *semi-Markov decision process* (SMDP) is given by a tuple $M = (S, \tau, \rho, L)$ where

- S is a non-empty, countable set of *states*,
- $\tau : S \times A \to \mathcal{D}(S)$ is the *transition function*,
- $\rho : S \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is the *residence-time function*, and
- $L : S \to 2^{\mathcal{AP}}$ is the *labelling function*. ▲

The operational behaviour of an SMDP is as follows. The SMDP at a given state $s \in S$, after receiving an input $a \in A$, goes to state $s' \in S$ within time t with probability $\tau(s,a)(s') \cdot \rho(s)([0,t])$. An SMDP is said to be *finite* if it has a finite set of states, and it is said to be *finitely supported* if its transition function $\tau(s,a)$ is finitely supported for every s and a. For $s \in S$, we will write F_s for the CDF of $\rho(s)$, i.e. $F_s(t) = \rho(s)([0,t])$.

Continuous-time Markov decision processes are a special case of SMDPs in which all residence-time functions are exponentially distributed, whereas discrete-time Markov decision processes are a special case of SMDPs where the residence-time distribution in each state is the Dirac measure at 0, representing the fact that transitions are taken instantaneously.

In defining simulation and bisimulation for SMDPs, we will use ingredients from the definition of simulation and bisimulation for Markov decision processes [23] and simulation and bisimulation for continuous-time Markov chains [3]. However, since we are also generalising to arbitrary distributions on time rather than just exponential distributions, the condition on rates for exponential distributions must be replaced with a more general condition on the distributions. There is a rich literature on so-called stochastic orders [24], which impose an ordering on random variables. We will consider here the most commonly used of these, known as the *usual stochastic order*.

Definition D.2.2. For an SMDP $M = (S, \tau, \rho, L)$, a relation $R \subseteq S \times S$ is a *simulation* (resp. *bisimulation*) on M if for all $(s_1, s_2) \in R$ we have

1. $L(s_1) = L(s_2)$,
2. $F_{s_2}(t) \geq F_{s_1}(t)$ (resp. $F_{s_2}(t) = F_{s_1}(t)$) for all $t \in \mathbb{R}_{\geq 0}$, and
3. for all $a \in A$ there exists a *weight function* or *coupling* $\Delta_a \colon S \times S \to [0,1]$ between $\tau(s_1, a)$ and $\tau(s_2, a)$ such that
   (a) $\Delta_a(s, s') > 0$ implies $(s, s') \in R$,
   (b) $\tau(s_1, a)(s) = \sum_{s' \in S} \Delta_a(s, s')$ for all $s \in S$, and
   (c) $\tau(s_2, a)(s') = \sum_{s \in S} \Delta_a(s, s')$ for all $s' \in S$.

We say that s_2 *simulates* (resp. is *bisimilar to*) s_1, written $s_1 \precsim s_2$ (resp. $s_1 \sim s_2$), if there is a simulation (resp. bisimulation) relation containing (s_1, s_2). ▲

It is easy to show that the *similarity* relation $\precsim$ is the largest simulation relation, and analogously that the *bisimilarity* relation $\sim$ is the largest bisimulation relation. The coupling ensures that the simulation relation is preserved by successor states. Intuitively, s_1 simulates s_2 if the CDF of $\rho(s_2)$ is pointwise greater than or equal to the CDF of $\rho(s_1)$, and the transition probability distribution of s_1 can be matched by the transition probability function s_2 by means of a coupling, in such a way that if two successor states s_1' and s_2' have

a non-zero coupling, then s_1' again simulates s_2'. For bisimulation, we instead require that the CDFs behave exactly the same in each step.

Given a set $C \subseteq S$ and a relation $R \subseteq S \times S$, let

$$R(C) = \{s' \in S \mid (s, s') \in R \text{ for some } s \in C\}$$

be the R-closure of C. If R is a preorder, $R(C)$ is the upward closure of C.

The following result, which is a trivial generalisation of [22, Lemma 1], gives a different but equivalent definition of simulation which is sometimes useful.

Proposition D.2.3. *$R \subseteq S \times S$ is a simulation relation if and only if for any $(s_1, s_2) \in R$ we have*

1. *$L(s_1) = L(s_2)$,*
2. *$F_{s_2}(t) \geq F_{s_1}(t)$ for all $t \in \mathbb{R}_{\geq 0}$, and*
3. *$\tau(s_1, a)(C) \leq \tau(s_2, a)(R(C))$ for all $C \subseteq S$.*

The following generalises [3, Proposition 25(3)] to the case of SMDPs.

Proposition D.2.4. $\precsim \cap \precsim^{-1} = \sim$.

Proof. First assume that $s_1 \sim s_2$. Then $L(s_1) = L(s_2)$. Also, $F_{s_2}(t) = F_{s_2}(t)$ for all t, so clearly $F_{s_2}(t) \geq F_{s_1}(t)$ and $F_{s_1}(t) \geq F_{s_2}(t)$ for all t. For any subset $C \subseteq S$ we have

$$\tau(s_1, a)(C) \leq \tau(s_1, a)(\sim(C)) = \tau(s_2, a)(\sim(C))$$

and

$$\tau(s_2, a)(C) \leq \tau(s_2, a)(\sim(C)) = \tau(s_1, a)(\sim(C)) \ ,$$

so $s_1 \precsim s_2$ and $s_1 \precsim^{-1} s_2$.

Now assume that $s_1 \precsim s_2$ and $s_1 \precsim^{-1} s_2$. Clearly $L(s_1) = L(s_2)$. Since $F_{s_2}(t) \geq F_{s_1}(t)$ and $F_{s_1}(t) \geq F_{s_2}(t)$ for all t, it follows that $F_{s_2}(t) = F_{s_1}(t)$ for all t. Now take an arbitrary subset $C \subseteq S$ and let $B = \precsim \cap \precsim^{-1}(C)$, $C_1 = \precsim(B)$, and $C_2 = C_1 \setminus B$. Then

$$\tau(s_1, a)(C_1) = \tau(s_1, a)(\precsim(C_1)) \leq \tau(s_2, a)(\precsim(C_1)) = \tau(s_2, a)(C_1)$$

and

$$\tau(s_2, a)(C_1) = \tau(s_2, a)(\precsim(C_1)) \leq \tau(s_1, a)(\precsim(C_1)) = \tau(s_1, a)(C_1) \ ,$$

so $\tau(s_1, a)(C_1) = \tau(s_2, a)(C_1)$, and likewise we can show that $\tau(s_1, a)(C_2) = \tau(s_2, a)(C_2)$. Since we can write

$$\tau(s_1, a)(C_1) = \tau(s_1, a)(B) + \tau(s_1, a)(C_2)$$

and

$$\tau(s_2, a)(C_1) = \tau(s_2, a)(B) + \tau(s_2, a)(C_2) \ ,$$

this together implies that $\tau(s_1, a)(B) = \tau(s_2, a)(B)$. Hence we conclude that $s_1 \sim s_2$. ■

The above is analogous to a result stating that bisimulation and simulation equivalence coincide for deterministic labelled transition systems [2]. In our case, Proposition D.2.4 holds because reactive systems are inherently deterministic.

D.3 Comparing the Speed of Residence-Time Distributions

For comparing the random variables describing the residence time on states, the similarity relation uses the usual stochastic order: if $s_1 \precsim s_2$ then, for all $t \in \mathbb{R}_{\geq 0}$, $F_{s_1}(t) \leq F_{s_2}(t)$. In words, if s_2 simulates s_1, it is more likely that s_2 will take a transition before s_1, that is, s_2 is stochastically faster than s_1 in reacting to an input.

In this section, we propose a different way of comparing residence-time distributions. The idea is to get quantitative information about how much a distribution should be accelerated to become at least as fast as another distribution.

Definition D.3.1. Let F and G be CDFs and $\varepsilon \in \mathbb{R}_{>0}$. We say that F is *ε-faster than G*, written $F \sqsubseteq_\varepsilon G$, if for all t we have $F(\varepsilon \cdot t) \geq G(t)$. ▲

Consider two states s_1 and s_2, having residence time governed by the distributions F_{s_1} and F_{s_2}, respectively, and assume that $F_{s_1} \sqsubseteq_\varepsilon F_{s_2}$ holds. If $0 < \varepsilon \leq 1$, then this means that s_1 is stochastically faster than s_2, even if the residence time in s_1 is slowed down by a factor ε. If instead we have $\varepsilon > 1$, then s_1 is stochastically slower than s_2, but if we accelerate its residence-time distribution by a factor ε, then it becomes stochastically faster than s_2.

In the rest of the section we will argue that $\sqsubseteq_\varepsilon$ is a good notion for gathering quantitative information about the speed of residence-time distributions on states. We will do this by comparing the most common distributions used in the literature for modelling residence time on states on stochastic systems: Dirac distributions, exponential distributions, and uniform distributions.

The Dirac measure at zero is the fastest measure, in the following sense.

Proposition D.3.2. *Let F be any CDF. The following holds for any $\varepsilon \in \mathbb{R}_{>0}$.*

1. *$Dirac[0] \sqsubseteq_\varepsilon F$.*
2. *If $F \neq Dirac[0]$, then $F \not\sqsubseteq_\varepsilon Dirac[0]$.*

Proof. The first point is clear, since $Dirac[0]\,(t) = 1 \geq F(t)$ for any t.

For the second point, note that $Dirac[0]$ is the only CDF such that

$$Dirac[0]\,(0) = 1,$$

and hence $F(\varepsilon \cdot 0) < Dirac[0]\,(0)$ for any $\varepsilon \geq 1$. ■

For comparing exponential distributions, it is simple to show that it is enough to accelerate by the ratio between the two rates. The same is true for uniform distributions, except we also need to consider whether the uniform distributions start at 0, since if a uniform distribution starts at 0, then we can only hope to make another uniform distribution faster than it if this other uniform distribution also starts at 0. To prove this, we make use of the following two lemmas.

Lemma D.3.3. *For $\varepsilon \in \mathbb{R}_{>0}$ it holds that $Exp[\theta]\,(\varepsilon \cdot t) = Exp[\varepsilon \cdot \theta]\,(t)$.*

Proof.

$$Exp[\theta]\,(\varepsilon \cdot t) = 1 - e^{-\theta \cdot (\varepsilon \cdot t)} = 1 - e^{-(\theta \cdot \varepsilon) \cdot t} = Exp[\varepsilon \cdot \theta]\,(t).$$ ■

Lemma D.3.4. *For $\varepsilon \in \mathbb{R}_{>0}$ it holds that $Unif[a,b]\,(\varepsilon \cdot t) = Unif\left[\frac{a}{\varepsilon}, \frac{b}{\varepsilon}\right]\,(t)$.*

Proof.

$$\frac{\varepsilon \cdot t - a}{b - a} = 0 \implies \varepsilon \cdot t - a = 0 \implies t = \frac{a}{\varepsilon}$$

and

$$\frac{\varepsilon \cdot t - a}{b - a} = 1 \implies \varepsilon \cdot t - a = b - a \implies t = \frac{b}{\varepsilon}.$$

Hence $Unif[a,b]\,(\varepsilon \cdot t) = Unif\left[\frac{a}{\varepsilon}, \frac{b}{\varepsilon}\right]\,(t)$. ■

Proposition D.3.5.

1. *$Exp[\theta_1] \sqsubseteq_\varepsilon Exp[\theta_2]$, where $\varepsilon = \frac{\theta_2}{\theta_1}$.*
2. *If $c = 0$ and $a > 0$, then $Unif[a,b] \not\sqsubseteq_\varepsilon Unif[c,d]$ for any $\varepsilon \in \mathbb{R}_{>0}$.*
3. *If $c = 0$ and $a = 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Unif[c,d]$, where $\varepsilon = \frac{b}{d}$.*
4. *If $c > 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Unif[c,d]$, where $\varepsilon = \max\left\{\frac{a}{c}, \frac{b}{d}\right\}$.*

In all cases, the given ε is the least such that the ε-faster than relation holds.

Proof. 1. We see that

$$Exp[\theta_1]\,(\varepsilon \cdot t) = 1 - e^{-\theta_1 \varepsilon \cdot t} = 1 - e^{-\theta_2 t} = Exp[\theta_2]\,(t).$$

If $\varepsilon' < \varepsilon$, then take some $t > 0$ to get

$$Exp[\theta_1]\,(\varepsilon' \cdot t) = 1 - e^{-\theta_1 \varepsilon' \cdot t} < 1 - e^{-\theta_1 \varepsilon \cdot t} = Exp[\theta_2]\,(t),$$

and hence $Exp[\theta_1] \not\sqsubseteq_{\varepsilon'} Exp[\theta_2]$.

2. Let $c = 0$ and $a > 0$. Take an arbitrary $\varepsilon \in \mathbb{R}_{>0}$ and let $t = \frac{a}{\varepsilon} > 0$ in order to get $Unif[a,b]\,(\varepsilon \cdot t) = Unif[a,b]\,(a) = 0$. However, $Unif[c,d]\,(t) > 0$ for any $t > 0$, so $Unif[a,b]\,(\varepsilon \cdot t) < Unif[c,d]\,(t)$.

3. Let $c = 0$ and $a = 0$, and take $\varepsilon = \frac{b}{d}$. Then

$$Unif[a,b]\,(\varepsilon \cdot t) = Unif\left[a \cdot \frac{d}{b}, d\right](t) = Unif[0,d]\,(t) = Unif[c,d]\,(t).$$

To show that it is the least ε such that the ε-faster-than relation holds, let $\varepsilon' < \frac{b}{d}$. Then

$$Unif[a,b]\,(\varepsilon' \cdot d) < Unif[a,b]\left(\frac{b}{d} \cdot d\right) = 1 = Unif[c,d]\,(d).$$

4. Now let $c > 0$ and $\varepsilon = \max\{\frac{a}{c}, \frac{b}{d}\}$. If $\max\{\frac{a}{c}, \frac{b}{d}\} = \frac{a}{c}$, then

$$Unif[a,b]\,(\varepsilon \cdot t) = Unif\left[c, b \cdot \frac{c}{a}\right](t).$$

Since $\frac{a}{c} \geq \frac{b}{d}$ we get $\frac{c}{a} \leq \frac{d}{b}$, and hence

$$Unif\left[c, b \cdot \frac{c}{a}\right](t) \geq Unif\left[c, b \cdot \frac{d}{b}\right](t) = Unif[c,d]\,(t).$$

On the other hand, if $\max\{\frac{a}{c}, \frac{b}{d}\} = \frac{b}{d}$, then we get $\frac{c}{a} \geq \frac{d}{b}$, and hence

$$Unif[a,b]\,(\varepsilon \cdot t) = Unif\left[a \cdot \frac{d}{b}, d\right](t) \geq Unif[c,d]\,(t).$$

It remains to prove that this is the least ε such that this relation holds. Let $\varepsilon' < \max\{\frac{a}{c}, \frac{b}{d}\}$. If $\max\{\frac{a}{c}, \frac{b}{d}\} = \frac{a}{c}$, then let $t = \frac{a}{\varepsilon'} > \frac{a}{\frac{a}{c}} = c$. Then $Unif[a,b]\,(\varepsilon' \cdot t) = Unif[a,b]\,(a) = 0$, but $Unif[c,d]\,(t) > 0$ since $t > c$. On the other hand, if $\max\{\frac{a}{c}, \frac{b}{d}\} = \frac{b}{d}$, then

$$Unif[a,b]\,(\varepsilon' \cdot d) < Unif[a,b]\left(\frac{b}{d} \cdot d\right) = 1 = Unif[c,d]\,(d). \qquad \blacksquare$$

Moreover, an exponential distribution can never be made faster than a uniform distribution, since uniform distributions become 1 eventually, but exponential distributions tend asymptotically to 1 but never reach it. Furthermore, whether or not a uniform distribution can be made faster than an exponential distribution depends on its value at 0.

Proposition D.3.6.

1. *$Exp[\theta] \not\sqsubseteq_\varepsilon Unif[a,b]$ for all $\varepsilon \in \mathbb{R}_{>0}$.*
2. *If $a > 0$, then $Unif[a,b] \not\sqsubseteq_\varepsilon Exp[\theta]$ for all $\varepsilon \in \mathbb{R}_{>0}$.*
3. *If $a = 0$, then $Unif[a,b] \sqsubseteq_\varepsilon Exp[\theta]$, where $\varepsilon = \theta \cdot b$. Furthermore, this is the least ε such that the ε-faster-than relation holds.*

Proof.

1. We have $Unif[a,b](b) = 1$, but $Exp[\theta](t) < 1$ for all t, and hence $Exp[\theta] \not\sqsubseteq_\varepsilon Unif[a,b]$ for any $\varepsilon \in \mathbb{R}_{>0}$.
2. If $a > 0$, then let $\varepsilon \in \mathbb{R}_{>0}$ be given, and let $t = \frac{a}{\varepsilon}$. Then $Unif[a,b](\varepsilon \cdot t) = Unif[a,b](a) = 0$, but $Exp[\theta](t) > 0$ since $t > 0$, and therefore $Unif[a,b] \not\sqsubseteq_\varepsilon Exp[\theta]$.
3. If $a = 0$, then let $\varepsilon = \theta \cdot b$. Clearly $Unif[a,b](\varepsilon \cdot 0) = 0$ and $Exp[\theta](0) = 0$. We see that $Unif[a,b](\varepsilon \cdot t) = \frac{\theta b t}{b}$ and $Exp[\theta](t) = 1 - e^{-\theta t}$ have the same derivative from the right at 0, namely θ. Hence the slope of these two functions is the same in 0, but since the slope of an exponential distribution is always decreasing, this means that $Unif[a,b] \sqsubseteq_\varepsilon Exp[\theta]$. If $\varepsilon' < \theta \cdot b$, then the slope in 0 of $Unif[a,b](\varepsilon' \cdot t)$ must be less than that of $Exp[\theta]$, so there will exist some $t > 0$ such that $Unif[a,b](\varepsilon' \cdot t) < Exp[\theta](t)$, and hence $Unif[a,b] \not\sqsubseteq_{\varepsilon'} Exp[\theta]$. ■

The ε-faster-than relation enjoys a kind of monotonicity property, which is simply a consequence of the fact that CDFs are increasing.

Lemma D.3.7. *Let $\varepsilon \leq \varepsilon'$ and assume that $F \sqsubseteq_\varepsilon G$. Then $F \sqsubseteq_{\varepsilon'} G$.*

Proof. $F \sqsubseteq_\varepsilon G$ means that $F(\varepsilon \cdot t) \geq G(t)$ for all t. Since F is non-decreasing and $\varepsilon \leq \varepsilon'$, this means that $F(\varepsilon' \cdot t) \geq F(\varepsilon \cdot t) \geq G(t)$ for all t, so $F \sqsubseteq_{\varepsilon'} G$. ■

The probability distribution of the sum of two independent random variables is the *convolution* of their individual distributions. The general formula for the convolution of two measures μ and ν on the real line is given by

$$(\mu * \nu)(E) = \int_0^\infty \nu(E - x)\, \mu(\mathrm{d}x).$$

Notably, the ε-faster-than relation is a congruence with respect to convolution of measures.

Proposition D.3.8. *If $F_{\mu_1} \sqsubseteq_\varepsilon F_{\mu_2}$ and $F_{\nu_1} \sqsubseteq_\varepsilon F_{\nu_2}$, then $F_{(\mu_1 * \nu_1)} \sqsubseteq_\varepsilon F_{(\mu_2 * \nu_2)}$.*

Proof. Define the transformation $T(x) = \varepsilon \cdot x$ and let $\nu_1'([0,t]) = \nu_1\left(\left[0, \frac{t}{\varepsilon}\right]\right)$. Then we see that

$$\begin{aligned}
\nu_1(T^{-1}([0,t])) &= \nu_1(\{x \mid x \cdot \varepsilon \in [0,t]\}) \\
&= \nu_1\left(\{x \mid x \in [0, t/\varepsilon]\}\right) \\
&= \nu_1\left(\left[0, \frac{t}{\varepsilon}\right]\right) \\
&= \nu_1'([0,t]).
\end{aligned}$$

Because $F_{\mu_1} \sqsubseteq_\varepsilon F_{\mu_2}$ we know that $\mu_1([0, \varepsilon \cdot t]) \geq \mu_2([0,t])$ for all t, and since $F_{\nu_1} \sqsubseteq_\varepsilon F_{\nu_2}$, we know that $\nu_1([0, \varepsilon \cdot t) = \nu_1'([0,t]) \geq \nu_2([0,t])$ for all t. We can therefore do following series of transformations [19, Proposition 3.8].

$$\begin{aligned}
(\mu_1 * \nu_1)([0, \varepsilon \cdot t]) &= \int_0^\infty \mu_1([0, \varepsilon \cdot t - x])\ \nu_1(\mathrm{d}x) \\
&= \int_0^\infty \mu_1([0, \varepsilon \cdot t - T(x)])\ \nu_1'(\mathrm{d}x) \\
&= \int_0^\infty \mu_1([0, \varepsilon(t - x))\ \nu_1'(\mathrm{d}x) \\
&\geq \int_0^\infty \mu_2([0, t - x])\ \nu_1'(\mathrm{d}x) \\
&\geq \int_0^\infty \mu_2([0, t - x])\ \nu_2(\mathrm{d}x) \\
&= (\mu_2 * \nu_2)([0,t]).
\end{aligned}$$

■

In Section D.7.1 we will see that the above property is essential for the preservation of reachability properties. Intuitively, this is because convolution corresponds to sequential composition of the residence-time behaviour.

There are other possible ways to compare the relative speed of residence-time distributions quantitatively. In the following we explore some alternative definitions of the notion of the ε-faster-than relation, and argue that none of them are preferable to the one given in Definition D.3.1. Given two CDFs F and G, we consider the following three alternative definitions of $F \sqsubseteq_\varepsilon G$:

1. for all t, $F(t) \cdot \varepsilon \geq G(t)$,
2. for all t, $F(t) + \varepsilon \geq G(t)$, and
3. for all t, $F(\varepsilon + t) \geq G(t)$.

If $\sqsubseteq_\varepsilon$ is defined as in (1), then we see that $\mathit{Unif}[a,b] \not\sqsubseteq_\varepsilon \mathit{Unif}[c,d]$, for any $\varepsilon \in \mathbb{R}_{>0}$ whenever $c < a$. This is because $\mathit{Unif}[a,b]\,(a) \cdot \varepsilon = 0 < \mathit{Unif}[c,d]\,(a)$. Hence we lose the properties of Proposition D.3.5.

If $\sqsubseteq_\varepsilon$ is defined as in (2), we trivially get that whenever $\varepsilon \geq 1$, $F \sqsubseteq_\varepsilon G$, for any two CDFs F and G. Hence (2) is only interesting for $0 \leq \varepsilon < 1$. However, even in this case we would still lose the properties of Proposition D.3.5. Indeed, whenever $a \geq d$, $\mathit{Unif}[a,b] \not\sqsubseteq_\varepsilon \mathit{Unif}[c,d]$, for any $0 \leq \varepsilon < 1$. This follows because $\mathit{Unif}[a,b](a) + \varepsilon = \varepsilon < 1 = \mathit{Unif}[c,d](a)$.

Lastly, if $\sqsubseteq_\varepsilon$ is defined as in (3), then it would not be a congruence with respect to convolution of distributions, i.e., Proposition D.3.8 would not hold. For a counterexample, take $F_{\mu_1} = \mathit{Unif}[2,4]$, $F_{\mu_2} = \mathit{Unif}[1,3]$, $F_{\nu_1} = \mathit{Unif}[3,4]$, and $F_{\nu_2} = \mathit{Unif}[2,4]$. Then $F_{\mu_1} \sqsubseteq_1 F_{\mu_2}$ and $F_{\nu_1} \sqsubseteq_1 F_{\nu_2}$, but $F_{(\mu_1 * \mu_2)} \not\sqsubseteq_1 F_{(\nu_1 * \nu_2)}$.

D.4 A Hemimetric for Semi-Markov Decision Processes

In this section, we are going to extend the definition of simulation relation between SMDPs to the quantitative setting. We will see that this relation naturally induces a notion of distance between SMDPs, describing the least acceleration factor required globally on the residence-time distributions to make a given SMDP as fast as another one.

Definition D.4.1. Let $\varepsilon \in \mathbb{R}_{>0}$. For an SMDP $M = (S, \tau, \rho, L)$, a relation $R \subseteq S \times S$ is an *ε-simulation relation* on M if for all $(s_1, s_2) \in R$ we have that the first and third condition for simulation are satisfied, and $F_{s_2} \sqsubseteq_\varepsilon F_{s_1}$. We say that s_2 *ε-simulates* s_1, written $s_1 \precsim_\varepsilon s_2$, if there is an ε-simulation relation R such that $(s_1, s_2) \in R$. ▲

Example D.4.2. Let $A = \{a\}$ and consider the SMDP $M = (S, \tau, \rho, L)$ given by $S = \{s_1, s_2\}$, $\tau(s_1, a)(s_1) = 1 = \tau(s_2, a)(s_2)$, $F_{s_1} = \mathit{Exp}[4]$, $F_{s_2} = \mathit{Exp}[2]$, and $L(s_1) = L(s_2)$. By Proposition D.3.5 we see that $s_1 \precsim_2 s_2$ and $s_2 \precsim_{\frac{1}{2}} s_1$. ♦

It is easy to show that the *ε-similarity* relation $\precsim_\varepsilon$ is the largest simulation relation, and with the previous section in mind, one immediately sees that $\precsim_1$ coincides with $\precsim$. Moreover, the following holds.

Proposition D.4.3. *For any $\varepsilon \leq 1$, if $s_1 \precsim_\varepsilon s_2$, then $s_1 \precsim s_2$.*

Proof. Let $R \subseteq S \times S$ be an ε-simulation relation such that $(s_1, s_2) \in R$. We will now argue that R is also a simulation relation. The first condition is clear. For the second condition, we have

$$F_{s_2}(t) \geq F_{s_2}(\varepsilon \cdot t) \geq F_{s_1}(t).$$

The third condition is satisfied because R is an ε-simulation relation. ■

If $\varepsilon > 1$, the above implication does not hold. For an easy counterexample consider s_1 and s_2 from Example D.4.2 where $s_1 \precsim_2 s_2$ but $s_1 \not\precsim s_2$.

For $\varepsilon > 1$, we can obtain a result similar to Proposition D.4.3 only if we "accelerate" the overall behaviour of s_2. Formally, for a given SMDP $M = (S, \tau, \rho, L)$, we define the SMDP $M_\varepsilon = (S_\varepsilon, \tau_\varepsilon, \rho_\varepsilon, L_\varepsilon)$ as follows:

$$\begin{aligned} S_\varepsilon &= S \cup \{(s)_\varepsilon \mid s \in S\}, \\ L_\varepsilon(s) &= L(s), \\ L_\varepsilon((s)_\varepsilon) &= L(s), \\ \rho_\varepsilon(s)([0,t]) &= \rho(s)([0,t]), \\ \rho_\varepsilon((s)_\varepsilon)([0,t]) &= \rho(s)([0,\varepsilon \cdot t]), \end{aligned} \qquad \begin{aligned} \tau_\varepsilon(s,a)(s') &= \tau(s,a)(s'), \\ \tau_\varepsilon(s,a)((s')_\varepsilon) &= 0, \\ \tau_\varepsilon((s)_\varepsilon,a)(s') &= 0, \\ \tau_\varepsilon((s)_\varepsilon,a)((s')_\varepsilon) &= \tau(s,a)(s'). \end{aligned}$$

Intuitively, the states $s \in S$ in M_ε are identical copies of those in M, whereas the states $(s)_\varepsilon$ react to each input $a \in A$ functionally identically to s but faster, since the residence-time on the states are all equally accelerated by a factor ε, thus $(s)_\varepsilon \precsim_\varepsilon s$. For this reason $(s)_\varepsilon$ is called the *ε-acceleration of s*.

Given the definition of accelerated state, Proposition D.4.3 can be generalised to arbitrary values of $\varepsilon \in \mathbb{R}_{>0}$ in the following way.

Proposition D.4.4. *For any $\varepsilon \in \mathbb{R}_{>0}$, $s_1 \precsim_\varepsilon s_2$ if and only if $s_1 \precsim (s_2)_\varepsilon$.*

Proof. ($\implies$) Let $R \subseteq S \times S$ be an ε-simulation relation with $(s_1, s_2) \in R$. Define

$$R' = \{(s, (s')_\varepsilon \in S_\varepsilon \times S_\varepsilon \mid (s, s') \in R\},$$

and take an arbitrary $(s, (s')_\varepsilon) \in R'$. The first condition of Definition D.4.1 is satisfied because $F_{(s')_\varepsilon}(t) = F_{s'}(\varepsilon \cdot t) \geq F_s(t)$.

For the second condition, we know that for any $a \in A$ there exists a coupling Δ_a, and we now define

$$\Delta'_a(s'', s''') = \begin{cases} 0 & \text{if } s'' \notin S \text{ or } s''' \in S \\ \Delta_a(s'', s''') & \text{otherwise.} \end{cases}$$

Since

$$\begin{aligned} \Delta'_a(s'', (s''')_\varepsilon) > 0 &\implies \Delta_a(s'', s''') > 0 \\ &\implies (s'', s''') \in R \\ &\implies (s'', (s''')_\varepsilon) \in R', \end{aligned}$$

condition (a) is also satisfied. For condition (b), first consider the case where $s'' \in S$. Then we get

$$\begin{aligned} \sum_{s''' \in S_\varepsilon} \Delta'_a(s'', s''') &= \sum_{(s''')_\varepsilon \in S_\varepsilon} \Delta'_a(s'', (s''')_\varepsilon) \\ &= \sum_{s''' \in S} \Delta_a(s'', s''') \\ &= \tau(s,a)(s'') \\ &= \tau_\varepsilon(s,a)(s''). \end{aligned}$$

For the case where $s'' \notin S$ we get

$$\sum_{s''' \in S_\varepsilon} \Delta'_a(s'', s''') = 0$$

and

$$\tau_\varepsilon(s, a)(s'') = 0.$$

Likewise, for condition (c), first consider the case where $s''' \in S$. Then

$$\sum_{s'' \in S_\varepsilon} \Delta'_a(s'', s''') = 0$$

and

$$\tau_\varepsilon((s')_\varepsilon, a)(s''') = 0.$$

For the case where $s''' \notin S$, we get

$$\begin{aligned}
\sum_{s'' \in S_\varepsilon} \Delta'_a(s'', s''') &= \sum_{s'' \in S} \Delta'_a(s'', s''') \\
&= \sum_{s'' \in S} \Delta_a(s'', s''') \\
&= \tau(s', a)(s''') \\
&= \tau_\varepsilon((s')_\varepsilon, a)((s''')_\varepsilon).
\end{aligned}$$

($\Longleftarrow$) Let $R \subseteq S_\varepsilon \times S_\varepsilon$ be a simulation relation with $(s_1, (s_2)_\varepsilon) \in R$ and define

$$R' = \{(s, s') \in S \times S \mid (s, (s')_\varepsilon) \in R\}.$$

For an arbitrary $(s, s') \in R'$ we get $F_{s'}(\varepsilon \cdot t) = F_{(s')_\varepsilon} \geq F_{s_1}(t)$, thus satisfying the first condition.

We know that for any $a \in A$ there exists a coupling Δ_a, and we now define

$$\Delta'_a(s'', s''') = \Delta_a(s'', (s''')_\varepsilon).$$

This coupling satisfies condition (a) because

$$\begin{aligned}
\Delta'_a(s'', s''') > 0 &\implies \Delta_a(s'', (s''')_\varepsilon) > 0 \\
&\implies (s'', (s''')_\varepsilon) \in R \\
&\implies (s'', s''') \in R'.
\end{aligned}$$

For condition (b), we see that

$$\begin{aligned}
\sum_{s''' \in S} \Delta'_a(s'', s''') &= \sum_{(s''')_\varepsilon \in S_\varepsilon} \Delta_a(s'', (s''')_\varepsilon) \\
&= \tau_\varepsilon(s, a)(s'') \\
&= \tau(s, a)(s'').
\end{aligned}$$

Likewise, for condition (c) we have

$$\begin{aligned}\sum_{s''\in S}\Delta'_a(s'',s''') &= \sum_{s''\in S}\Delta_a(s'',(s''')_\varepsilon)\\ &= \tau_\varepsilon((s')_\varepsilon,a)((s''')_\varepsilon)\\ &= \tau(s',a)(s''').\end{aligned}$$

■

The relevance of the above statement is twofold: it clarifies the relation between $\precsim_\varepsilon$ and $\precsim$, and also provides a way to modify the behaviour of a state s_2 of an SMDP in order to simulate a state s_1 whenever $s_1 \precsim_\varepsilon s_2$ holds.

Having this characterisation of similarity in terms of acceleration of processes one can think about the following problem: given two states, s_1 and s_2 such that $s_1 \not\precsim s_2$, what is the least $\varepsilon \geq 1$ (if it exists) such that $s_1 \precsim (s_2)_\varepsilon$ holds? We can answer this question by means of the following distance.

Definition D.4.5. The *simulation distance* $d\colon S \times S \to [1,\infty]$ between two states s_1 and s_2 is given by

$$d(s_1,s_2) = \inf\{\varepsilon \geq 1 \mid s_1 \precsim_\varepsilon s_2\}.$$

▲

As usual, if there is no $\varepsilon \geq 1$ such that $s_1 \precsim_\varepsilon s_2$, then $d(s_1,s_2) = \infty$, because $\inf\emptyset = \infty$. It is clear from the definition that $s_1 \precsim s_2$ implies $d(s_1,s_2) = 1$. For finitely supported SMDPs, the converse is also true. However, the proof of this makes use of the logical characterisation of the faster-than relation, so we delay the proof until Section D.7 where we discuss logical properties.

Note that the definition above does not give a distance in the usual sense, for two reasons: d is not symmetric and it does not satisfy the triangle inequality. One can show instead that d satisfies a multiplicative version of the triangle inequality, namely, that for all $s_1,s_2,s_3 \in S$, $d(s_1,s_3) \leq d(s_1,s_2) \cdot d(s_2,s_3)$. This is a direct consequence of the following properties of $\precsim_\varepsilon$. The first property states that $\precsim_\varepsilon$ is monotonic with respect to increasing values of ε.

Lemma D.4.6. *If $s_1 \precsim_\varepsilon s_2$ and $\varepsilon \leq \varepsilon'$, then $s_1 \precsim_{\varepsilon'} s_2$.*

Proof. This follows from Lemma D.3.7. ■

The second property is a quantitative generalisation of transitivity from which the multiplicative inequality discussed above follows.

Lemma D.4.7. *If $s_1 \precsim_\varepsilon s_2$ and $s_2 \precsim_{\varepsilon'} s_3$, then $s_1 \precsim_{\varepsilon\cdot\varepsilon'} s_3$.*

Proof. Since $s_1 \precsim_\varepsilon s_2$ and $s_2 \precsim_{\varepsilon'} s_3$, there exists an ε-simulation relation R such that $(s_1,s_2) \in R$ and an ε'-simulation relation R' such that $(s_2,s_3) \in R'$. First construct a relation

$$R'' = R \circ R' = \{(s'_1,s'_3) \in S \times S \mid \exists s'_2.((s'_1,s'_2) \in R \text{ and } (s'_2,s'_3) \in R')\}.$$

Now pick an arbitrary pair $(s_1', s_3') \in R''$. Clearly there exists s_2' such that $(s_1', s_2') \in R$ and $(s_2', s_3') \in R'$. Hence $L(s_1') = L(s_2') = L(s_3')$ and $F_{s_2'}(\varepsilon \cdot t) \geq F_{s_1'}(t)$ and $F_{s_3'}(\varepsilon' \cdot \varepsilon \cdot t) \geq F_{s_2'}(\varepsilon \cdot t)$, so $F_{s_3'}(\varepsilon' \cdot \varepsilon \cdot t) \geq F_{s_1'}(t)$, meaning $F_{s_3'} \sqsubseteq_{\varepsilon \cdot \varepsilon'} F_{s_1'}$. Thus the first and second conditions are satisfied.

Now let $a \in A$. There exists a coupling Δ_a between $\tau(s_1', a)$ and $\tau(s_2', a)$ and another coupling Δ_a' between $\tau(s_2', a)$ and $\tau(s_3', a)$. Next we construct a coupling between $\tau(s_1', a)$ and $\tau(s_3', a)$ by

$$\Delta_a''(s, s'') = \sum_{s' \in \mathrm{supp}(\tau(s_2', a))} \frac{\Delta_a(s, s') \cdot \Delta_a'(s', s'')}{\tau(s_2', a)(s')}. \tag{D.1}$$

We first verify that this is a coupling.

$$\begin{aligned}
\sum_{s'' \in S} \Delta_a''(s, s'') &= \sum_{s'' \in S} \sum_{s' \in \mathrm{supp}(\tau(s_2, a))} \frac{\Delta_a(s, s') \cdot \Delta_a'(s', s'')}{\tau(s_2, a)(s')} \\
&= \sum_{s' \in \mathrm{supp}(\tau(s_2, a))} \frac{\Delta_a(s, s') \cdot \sum_{s'' \in S} \Delta_a'(s', s'')}{\tau(s_2, a)(s')} \\
&= \sum_{s' \in \mathrm{supp}(\tau(s_2, a))} \frac{\Delta_a(s, s') \cdot \tau(s_2, a)(s')}{\tau(s_2, a)(s')} \\
&= \sum_{s' \in \mathrm{supp}(\tau(s_2, a))} \Delta_a(s, s') \\
&= \tau(s_1, a)(s),
\end{aligned}$$

and likewise we can show that $\sum_{s \in S} \Delta_a''(s, s'') = \tau(s_3, a)(s'')$. Now assume $\Delta_a''(s, s'') > 0$. By (D.1), this means that there must exist some

$$s' \in \mathrm{supp}(\tau(s_2', a)) \quad \text{such that} \quad \Delta_a(s, s') > 0 \text{ and } \Delta_a'(s', s'') > 0.$$

This implies that $(s, s') \in R$ and $(s', s'') \in R'$, so by the construction of R'' we get $(s, s'') \in R''$.

Hence we have shown that R'' is an $\varepsilon \cdot \varepsilon'$-simulation relation. Since clearly $(s_1, s_3) \in R''$, it follows that $s_1 \precsim_{\varepsilon \cdot \varepsilon'} s_3$. ■

Typically, one still uses the term distance for such multiplicative distances, because by applying the logarithm one does obtain a hemimetric.

Theorem D.4.8. $\log d$ *is a hemimetric.*

Proof. Let $d^{\log}(s_1, s_2) = \log d(s_1, s_2)$. Clearly, $d^{\log}(s_1, s_2) \geq 0$, and since $d(s, s) = 1$, $d^{\log}(s, s) = \log(1) = 0$. Hence it only remains to verify the triangle inequality.

If $d(s_1, s_2) \cdot d(s_2, s_3) = \infty$, then clearly $d(s_1, s_3) \leq d(s_1, s_2) \cdot d(s_2, s_3)$. If $d(s_1, s_2) \cdot d(s_2, s_3) \neq \infty$, then the sets $\{\varepsilon \geq 1 \mid s_1 \precsim_\varepsilon s_2\}$ and $\{\varepsilon' \geq 1 \mid s_2 \precsim_{\varepsilon'} s_3\}$

are both non-empty, so there must exist $\varepsilon, \varepsilon' \geq 1$ such that $s_1 \precsim_\varepsilon s_2$ and $s_2 \precsim_{\varepsilon'} s_3$, so by Lemma D.4.7 we have $s_1 \precsim_{\varepsilon\cdot\varepsilon'} s_3$, and hence $d(s_1, s_3) \neq \infty$. Taking the contrapositive of this, we get that $d(s_1, s_3) = \infty$ implies that $d(s_1, s_2) \cdot d(s_2, s_3) = \infty$, and hence also $d(s_1, s_3) \leq d(s_1, s_2) \cdot d(s_2, s_3)$.

Now assume that $d(s_1, s_3) \neq \infty$ and $d(s_1, s_2) \cdot d(s_2, s_3) \neq \infty$, and assume towards a contradiction that $d(s_1, s_3) > d(s_1, s_2) \cdot d(s_2, s_3)$. Since d is defined as an infimum, there must exist $\varepsilon, \varepsilon' \geq 1$ such that $s_1 \precsim_\varepsilon s_2$, $s_2 \precsim_{\varepsilon'} s_3$, and

$$d(s_1, s_3) > \varepsilon \cdot \varepsilon' \geq d(s_1, s_2) \cdot d(s_2, s_3).$$

However, by Lemma D.4.7 we have $\varepsilon \cdot \varepsilon' \geq d(s_1, s_3)$, a contradiction. Hence we get $d(s_1, s_3) \leq d(s_1, s_2) \cdot d(s_2, s_3)$, and by taking logarithms, we get

$$d^{\log}(s_1, s_3) \leq d^{\log}(s_1, s_2) + d^{\log}(s_2, s_3). \qquad \blacksquare$$

Example D.4.9. Consider again the SMDP from Example D.4.2. We can now see that $d(s_1, s_2) = 2$ and $d(s_2, s_1) = \frac{1}{2}$. This also shows that our distance is not symmetric, and hence not a pseudometric. ⧫

Remark D.4.10. From a topological point of view, there is no real difference between satisfying the standard triangle inequality or its multiplicative version. The difference essentially amounts to working either in the monoid $(\mathbb{R}_{\geq 0}, +)$ or the monoid $(\mathbb{R}_{\geq 1}, \cdot)$. However, these monoids are isomorphic via the bijection given by the logarithm and exponential functions. Since these functions are also continuous, the isomorphism is actually a homeomorphism, so all topological properties are preserved. ⧫

If one allowed $\varepsilon < 1$ in the distance, then one would get strange results such as a constant sequence which does not convergence to the constant element. To see this, consider a sequence $\{s_n\}_{n\in\mathbb{N}}$, where $s_n = s$ for any $n \in \mathbb{N}$. Then $d(s_n, s) = 1$ for any element of the sequence. However, $\lim_{k\to\infty} s_k \ni s$ means that for any $\varepsilon > 0$, there exists an $N \in \mathbb{N}$ such that $d(s_n, s) < \varepsilon$ for all $n > N$. However, this clearly does not hold for $0 < \varepsilon < 1$, so $\lim_{k\to\infty} s_k \not\ni s$.

D.5 Computing the Simulation Distance

In this section we provide an algorithm to compute the simulation distance given in Definition D.4.5 for finite SMDPs. The algorithm is shown to run in polynomial time for the distributions we have considered so far.

The following technical lemma will provide a sound basis for the correctness of the algorithm. Given two CDFs F and G, let

$$c(F, G) = \inf\{\varepsilon \geq 1 \mid F \sqsubseteq_\varepsilon G\}$$

denote the least acceleration factor needed by F to be faster than G. Given an SMDP M, we then define

$$\mathcal{C}(M) = \{c(F_{s'}, F_s) \mid s, s' \in S\}.$$

Lemma D.5.1. *Let M be a finite SMDP. If $d(s_1, s_2) \neq \infty$, then*

- *$s_1 \precsim_c s_2$, for some $c \in \mathcal{C}(M)$ and*
- *$d(s_1, s_2) = \min\{c \in \mathcal{C}(M) \mid s_1 \precsim_c s_2\}$.*

Proof. For the first claim, note that $d(s_1, s_2) \neq \infty$ implies that $s_1 \precsim_\varepsilon s_2$ for some $\varepsilon \geq 1$. This is witnessed by some ε-simulation relation which we denote by R. Now let

$$c^* = \max\{c \in \mathcal{C}(M) \mid c = c_{\rho(s'),\rho(s)} \text{ for some } (s, s') \in R\}.$$

Then it is clear that R is also a c^*-simulation relation, and hence $s_1 \precsim_{c^*} s_2$.

For the second claim, let

$$c_* = \min\{c \in \mathcal{C}(M) \mid s_1 \precsim_c s_2\}$$

and

$$X = \{\varepsilon \geq 1 \mid s_1 \precsim_\varepsilon s_2\}.$$

We first show that c_* is a lower bound of X. If $s_1 \precsim_\varepsilon s_2$, then by the previous argument we also have $s_1 \precsim_{c^*} s_2$. Note that $\varepsilon \geq c^*$, because otherwise we would have had $F_{s'} \not\sqsubseteq_\varepsilon F_s$ for some $(s, s') \in R$, contradicting the fact that R is a ε-simulation relation. Hence

$$\varepsilon \geq c^* \geq \min\{c \in \mathcal{C}(M) \mid s_1 \precsim_c s_2\} = c_*,$$

so c_* is a lower bound of X. Next we show that c_* is the greatest lower bound of X. We know that $s_1 \precsim_{c_*} s_2$, and hence $c_* \in X$, so if $\varepsilon > c_*$, then ε can not be a lower bound of X. Hence c_* is the greatest lower bound of X, and therefore we conclude that

$$c_* = \min\{c \in \mathcal{C}(M) \mid s_1 \precsim_c s_2\} = \inf X = d(s_1, s_2). \qquad \blacksquare$$

Lemma D.5.1 provides a strategy for computing the simulation distance between any two states s_1 and s_2 of a given SMDP M as follows. First, one constructs the set $\mathcal{C}(M)$. If $s_1 \precsim_c s_2$ does not hold for any $c \in \mathcal{C}(M)$, then the distance must be infinite; otherwise, it is the smallest $c \in \mathcal{C}(M)$ for which $s_1 \precsim_c s_2$ holds.

In order for this strategy to work, we need two ingredients: first, we should be able to compute the set $\mathcal{C}(M)$ and second, for any $c \in \mathcal{C}(M)$, we need an algorithm for checking whether $s_1 \precsim_c s_2$.

Recall that SMDPs allow for *arbitrary* residence-time distributions in each state. Therefore, it is not guaranteed that for any SMDP M the set $\mathcal{C}(M)$ can be computed. With the following definition we identify the class of SMDPs for which this can be done.

Definition D.5.2. A class $\mathcal{C}$ of CDFs is *effective* if for any $F, G \in \mathcal{C}$, $c(F, G)$ is computable. An SMDP M is effective if $\{F_s \mid s \in S\}$ is an effective class. ▲

In particular, for any pair of states s, s' of an effective SMDP M, we can decide whether $F_{s'} \sqsubseteq_\varepsilon F_s$ by simply checking whether $\varepsilon \geq c(F_{s'}, F_s)$. We will denote by $f(l)$ the complexity of computing $c(F_{s'}, F_s)$ for two arbitrary $s, s' \in S$ as a function of the length l of the representation of the residence-time distributions of M.

Let $\mathcal{C}_\Lambda$ denote the class consisting of the Dirac distribution at 0 as well as uniform and exponential distributions with rational parameters. By Propositions D.3.2–D.3.6 we immediately see that $\mathcal{C}_\Lambda$ is an effective class, and in fact it can be computed using only simple operations such as multiplication, division, and taking maximum. Hence $f(l)$ has constant complexity[1] whenever M takes residence-time distributions from $\mathcal{C}_\Lambda$.

Next we consider how to decide $s_1 \precsim_\varepsilon s_2$ for a given rational $\varepsilon \geq 1$. A decision procedure can be obtained by adapting to our setting the algorithm by Baier et al. [2] for deciding the simulation preorder between probabilistic labelled transition systems. The algorithm from [2] uses a partition refinement approach to compute the largest simulation relation and runs in time $\mathcal{O}(mn^7/\log n)$ for reactive systems, where $m = |A|$ is the number of actions, and $n = |S|$ is the number of states. Given $\varepsilon \geq 1$, we can proceed correspondingly to compute ε-similarity: we start from the relation $R = S \times S$ and update it by removing all the pairs (s, s') of states not satisfying the conditions of Definition D.4.1. This process is repeated on the resulting updated relation until no more pairs of states are removed. The resulting relation is the largest ε-simulation. Hence, checking $s_1 \precsim_\varepsilon s_2$ corresponds to determining whether the pair (s_1, s_2) is contained in the relation returned by the above algorithm. The complexity can be improved to $\mathcal{O}(m^2 n)$ by storing important information about the previous iteration of the algorithm and use this in the current iteration [27, Theorem 5.2.5].

Theorem D.5.3. *Let M be a finite and effective SMDP. Given $s_1, s_2 \in S$ and $\varepsilon \geq 1$, deciding whether $s_1 \precsim_\varepsilon s_2$ can be done in time $\mathcal{O}(n^2(f(l) + k) + m^2 n)$, where $k = |AP|$ is the number of atomic propositions.*

Proof. The algorithm for deciding $s_1 \precsim_\varepsilon s_2$ is essentially the same as that for deciding untimed simulation. Since we have assumed effectiveness, when

[1] As is standard, we consider numbers to be represented as floating points of bounded size in their binary representation.

choosing whether to remove a pair (s, s') from the current relation, we can check the conditions on Definition D.4.1. However, we also need to check whether $L(s) = L(s')$. For this we assume that the set of atomic propositions $\mathcal{AP}$ have an ordering $\mathcal{AP} = x_0, x_1, \ldots$, and that $L(s)$ is represented as a binary array where the ith entry in the array is 1 if $x_i \in L(s)$, and 0 otherwise. Then checking whether $L(s) = L(s')$ amounts to checking whether each array has the same entries, which can be done in time $k = |\mathcal{AP}|$. Testing for the existence of a coupling can be done in time $\mathcal{O}(m^2 n)$ by using the algorithm from [27], Hence we get a time complexity of $\mathcal{O}(n^2(f(l) + k) + m^2 n)$. ■

The algorithm for computing the simulation distance is given in Algorithm D.5.1. The algorithm starts by ordering the elements of $\mathcal{C}(M)$ as $c_1, \ldots, c_n$ while removing ∞ from the list. Then it searches for the smallest c_i such that $s_1 \precsim_{c_i} s_2$ holds. This is done by means of a bisection method. If $s_1 \precsim_{c_1} s_2$ holds, then c_1 is the smallest element such that this holds, so we return it. If $s_1 \precsim_{c_n} s_2$ does not hold, then, by Lemma D.4.6, $s_1 \precsim_{c_i} s_2$ does not hold for any $1 \leq i \leq n$, so we return ∞. If none of the above apply, at this point of the algorithm (line 4) we have that $s_1 \not\precsim_{c_1} s_2$ and $s_1 \precsim_{c_n} s_2$.

We use the variables i and j, respectively, as the left and right endpoints of the bisection interval. The bisection interval keeps track of those elements c_n for which we still do not know whether $s_1 \precsim_{c_n} s_2$. At the beginning, $i = 1$ and $j = n$. At line 7, $h = \left\lceil \frac{j-i}{2} \right\rceil$ is used as the decrement factor for the length of the bisection interval at each step. Since $h > 0$, the bisection interval decreases in size for each iteration. If $s_1 \precsim_{c_{j-h}} s_2$ holds, then $j - h$ is the current smallest element in $\mathcal{C}(M)$ for which this holds, hence $j - h$ will become the new right endpoint of the interval; otherwise $i + h$ is the new left endpoint. The bisection method stops when the endpoints meet or cross each other, at which point we know that $s_1 \not\precsim_{c_n} s_2$ for all $n < j$ and $s_1 \precsim_{c_n} s_2$ for all $n \geq j$, and hence we return c_j.

Computing the set $\mathcal{C}(M)$ at line 1 has complexity $n^2 f(l)$, and sorting it can be done in time $\mathcal{O}(n \log n)$ using mergesort. By Theorem D.5.3, and since we have already computed $\mathcal{C}(M)$, each of the ε-simulation checks in lines 2, 3, and 8 can be done in time $\mathcal{O}(n^2 k + m^2 n)$, but the complexity $n^2 k$ from comparing labels only needs to be computed once. Since the bisection interval is halved each time, the while-loop is taken at most $\log n$ times. We therefore get an overall time complexity of $\mathcal{O}(n^2(f(l) + k) + m^2 n \cdot \log n)$.

Theorem D.5.4. *Let M be a finite and effective SMDP. The simulation distance between any two states can be computed in time $\mathcal{O}(n^2(f(l) + k) + m^2 n \cdot \log n)$.*

Proof. Consider the algorithm described in Algorithm D.5.1. The correctness of the algorithm is given by Lemma D.5.1. We will now argue that the algorithm runs in time $\mathcal{O}(n^2(f(l) + k) + m^2 n \cdot \log n)$. The sorting in line

```
1  Order the elements of C(M) \ {∞} such that c_1 < c_2 < ··· < c_n ;
2  if s_1 ≾_{c_1} s_2 then return c_1 ;
3  else if s_1 ⋠_{c_n} s_2 then return ∞ ;
4  else
5  |  i ← 1, j ← n ;
6  |  while i < j do
7  |  |  h ← ⌈(j−i)/2⌉ ;
8  |  |  if s_1 ≾_{c_{j−h}} s_2 then j ← j − h ;
9  |  |  else i ← i + h ;
10 |  end
11 |  return c_j ;
12 end
```

Algorithm D.5.1: Computing the simulation distance between s_1 and s_2.

1 of the algorithm can be done in time $\mathcal{O}(n \log n)$ using mergesort. The checks in line 2 and 3 each has complexity $\mathcal{O}(n^2(f(l)+k)+m^2n)$ by Theorem D.5.3. However, we only need to compare labels and residence-time distributions once and then we can store the results for future iterations. Hence the complexity $\mathcal{O}(n^2(f(l)+k))$ is only incurred once, and not in each iteration. The while loop halves the number of elements left to check for each iteration, and hence it will loop at most $\log n$ times. Since in each iteration we make the check in line 8, the complexity of the while loop becomes $\mathcal{O}(n^2(f(l)+k)+m^2n \cdot \log n)$. ■

D.6 Compositional Properties

One of the most successful principles of formal verification is the notion of compositional reasoning, in which a large system can be understood in terms of its smaller components [6]. However, for this principle to work, one must ensure that properties inferred about the components carry over to the full, composite system. For bisimulation, this means that one wants bisimulation to be a congruence with respect to parallel composition, and a precongruence in the case of simulation. A natural generalisation of this is that of non-expansiveness, which means that parallel composition does not increase (expand) the distance between states. In this section we will prove that some natural notions of parallel composition on SMDPs are non-expansive with respect to the simulation distance.

First we define what it means to compose two SMDPs in parallel. As argued in [25], the style of synchronous CSP is the one that is most suitable

for SMDPs, so this is the one we will adopt here.

Definition D.6.1. A function $\star : \mathcal{D}(\mathbb{R}_{\geq 0}) \times \mathcal{D}(\mathbb{R}_{\geq 0}) \to \mathcal{D}(\mathbb{R}_{\geq 0})$ is a *residence-time composition function* if it is commutative. ▲

Definition D.6.2. Let $\star$ be a residence-time composition function. Then the $\star$*-composition* of $M_1 = (S_1, \tau_1, \rho_1, L_1)$ and $M_2 = (S_2, \tau_2, \rho_2, L_2)$, denoted $M_1 \parallel_\star M_2 = (S, \tau, \rho, L)$, is given as follows, for arbitrary $s_1, s_1' \in S_1$, $s_2, s_2' \in S_2$, and $a \in A$.

1. $S = S_1 \times S_2$,
2. $\tau((s_1, s_2), a)((s_1', s_2')) = \tau_1(s_1, a)(s_1') \cdot \tau_2(s_2, a)(s_2')$,
3. $\rho((s_1, s_2)) = \star(\rho_1(s_1), \rho_2(s_2))$, and
4. $L((s_1, s_2)) = L(s_1) \cup L(s_2)$. ▲

Given a composite system $M_1 \parallel_\star M_2 = (S, \tau, \rho, L)$, we write $s_1 \parallel_\star s_2$ to mean $(s_1, s_2) \in S$. The residence-time composition function $\star$ allows us to accommodate many different ways of combining timing behaviour, including those found in the literature on process algebras. We recall here some of these.

Maximum composition: $F_{\star(\mu,\nu)}(t) = \max(F_\mu(t), F_\nu(t))$.

For exponential distributions, $F_\mu = Exp[\theta]$ and $F_\nu = Exp[\theta']$, the following alternatives can be found.

Product rate composition: $F_{\star(\mu,\nu)} = Exp[\theta \cdot \theta']$.

Minimum rate composition: $F_{\star(\mu,\nu)} = Exp[\min\{\theta, \theta'\}]$.

Maximum rate composition: $F_{\star(\mu,\nu)} = Exp[\max\{\theta, \theta'\}]$.

Maximum composition is used for interactive Markov chains [13], product rate composition is used in SPA [14], minimum rate composition is used in PEPA [15], and maximum rate composition is used in TIPP [11].

In order to have non-expansiveness for $\star$-composition of SMDPs, we will need to restrict to residence-time composition functions $\star$ that are monotonic.

Definition D.6.3. A residence-time composition function $\star$ is *monotonic* if for all $\varepsilon \geq 1$ and $\mu, \nu, \eta \in \mathcal{D}(\mathbb{R}_{\geq 0})$, it holds that

$$F_\mu \sqsubseteq_\varepsilon F_\nu \quad \text{implies} \quad F_{\star(\mu,\eta)} \sqsubseteq_\varepsilon F_{\star(\nu,\eta)}. \qquad \blacktriangle$$

Requiring monotonicity is not a significant restriction, as many of the composition functions that are found in the literature are indeed monotonic.

Lemma D.6.4. *Maximum composition as well as product, minimum, and maximum rate composition are all monotonic.*

Proof. Let $\varepsilon \geq 1$ and assume that $F_\mu(\varepsilon \cdot t) \geq F_\nu(t)$ for all t.

We first consider maximum composition. If $F_{\star(\mu,\eta)}(\varepsilon \cdot t) = F_\mu(\varepsilon \cdot t)$, then $F_\mu(\varepsilon \cdot t) \geq F_\eta(\varepsilon \cdot t) \geq F_\eta(t)$, so

$$F_{\star(\mu,\eta)}(\varepsilon \cdot t) = F_\mu(\varepsilon \cdot t) \geq F_{\star(\nu,\eta)}(t).$$

On the other hand, consider the case where $F_{\star(\mu,\eta)}(\varepsilon \cdot t) = F_\eta(\varepsilon \cdot t)$. Then we know that $F_{\star(\mu,\eta)}(\varepsilon \cdot t) \geq F_\mu(\varepsilon \cdot t) \geq F_\nu(t)$. If it is the case that $F_{\star(\nu,\eta)}(t) = F_\nu(t)$, then

$$F_{\star(\mu,\eta)}(\varepsilon \cdot t) \geq F_\nu(t) = F_{\star(\nu,\eta)}(t).$$

If $F_{\star(\nu,\eta)}(t) = F_\eta(t)$, then

$$F_{\star(\mu,\eta)}(\varepsilon \cdot t) = F_\eta(\varepsilon \cdot t) \geq F_\eta(t) = F_{\star(\nu,\eta)}(t).$$

So we conclude that $F_{\star(\mu,\eta)}(t) \geq F_{\star(\nu,\eta)}(t')$.

Next we consider the different rate compositions. Assume that $F_\mu = Exp[\theta]$, $F_\nu = Exp[\theta']$, and $F_\eta = Exp[\theta'']$. Since we have assumed $F_\mu(\varepsilon \cdot t) \geq F_\nu(t)$ for all t, this implies by Lemma D.3.3 that $\varepsilon \cdot \theta \geq \theta'$.

For product rate composition, note that $\varepsilon \cdot \theta \geq \theta'$ implies $\varepsilon \cdot \theta \cdot \theta'' \geq \theta' \cdot \theta''$. Therefore

$$\begin{aligned} F_{\star(\mu,\eta)}(\varepsilon \cdot t) &= Exp\left[\theta \cdot \theta''\right](\varepsilon \cdot t) = Exp\left[\varepsilon \cdot \theta \cdot \theta''\right](t) \\ &\geq Exp\left[\theta' \cdot \theta''\right](t) = F_{\star(\nu,\eta)}(t). \end{aligned}$$

For minimum rate composition, we want to show that $\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} \geq \min\{\theta', \theta''\}$. If $\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} = \varepsilon \cdot \theta$, then

$$\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} = \varepsilon \cdot \theta \geq \theta' \geq \min\{\theta', \theta''\}.$$

Otherwise, if $\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} = \varepsilon \cdot \theta''$, then

$$\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} = \varepsilon \cdot \theta'' \geq \theta'' \geq \min\{\theta', \theta''\}.$$

Hence

$$F_{\star(\mu,\eta)}(\varepsilon \cdot t) = Exp\left[\min\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\}\right](t) \geq Exp\left[\min\{\theta', \theta''\}\right](t) = F_{\star(\nu,\eta)}(t).$$

For maximum composition, we see that if $\max\{\theta', \theta''\} = \theta'$, then

$$\max\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} \geq \varepsilon \cdot \theta \geq \theta' = \max\{\theta', \theta''\},$$

and if $\max\{\theta', \theta''\} = \theta''$, then

$$\max\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\} \geq \varepsilon \cdot \theta'' \geq \theta'' = \max\{\theta', \theta''\}.$$

Hence

$$F_{\star(\mu,\eta)}(\varepsilon \cdot t) = Exp\big[\max\{\varepsilon \cdot \theta, \varepsilon \cdot \theta''\}\big](t) \geq Exp\big[\max\{\theta', \theta''\}\big](t) = F_{\star(\nu,\eta)}(t).$$

■

Now we can prove that the $\star$-composition of finite SMDPs is indeed non-expansive with respect to the simulation distance, provided that $\star$ is monotonic.

Theorem D.6.5. *For finite SMDPs and monotonic $\star$,*

$$d(s_1, s_2) \leq \varepsilon \quad \textit{implies} \quad d(s_1 \parallel_\star s_3, s_2 \parallel_\star s_3) \leq \varepsilon.$$

Proof. Assume that $d(s_1, s_2) \leq \varepsilon$. By Lemma D.5.1, we have that $s_1 \precsim_{d(s_1,s_2)} s_2$, so by Lemma D.4.6 we get that $s_1 \precsim_\varepsilon s_2$. Hence, there exists a ε-simulation relation R such that $(s_1, s_2) \in R$. Now construct

$$R' = \{(s_1' \parallel_\star s_3, s_2' \parallel_\star s_3) \mid (s_1', s_2') \in R \text{ and } s_3 \in S\},$$

and we want to show that R' is a ε-simulation relation.

Pick some $(s_1' \parallel_\star s_3, s_2' \parallel_\star s_3) \in R'$. Then we get

$$L(s_1' \parallel_\star s_3) = L(s_1') \cup L(s_3) = L(s_2') \cup L(s_3) = L(s_2' \parallel_\star s_3).$$

Since $\star$ is monotonic, we immediately get

$$\star(\rho(s_2'), \rho(s_3))([0, \varepsilon \cdot t]) \geq \star(\rho(s_1'), \rho(s_3))([0, t])$$

for all t, so $F_{s_2' \parallel_\star s_3} \sqsubseteq_\varepsilon F_{s_1' \parallel_\star s_3}$. Now let $a \in A$ be an arbitrary action and define

$$\Delta_a'(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3'') = \begin{cases} 0 & \text{if } s_3' \neq s_3'' \\ \Delta_a(s_1'', s_2'') \cdot \tau(s_3, a)(s_3') & \text{otherwise.} \end{cases}$$

If $\Delta_a'(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3'') > 0$, then $s_3' = s_3''$ and also $\Delta_a(s_1'', s_2'') > 0$, so $(s_1'', s_2'') \in R$, and hence $(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3'') \in R'$. Furthermore,

$$\begin{aligned} \sum_{s_2'' \parallel_\star s_3''} \Delta_a'(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3'') &= \sum_{s_2''} \Delta_a'(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3') \\ &= \sum_{s_2''} \Delta_a(s_1'', s_2'') \cdot \tau(s_3, a)(s_3') \\ &= \tau(s_3, a)(s_3') \cdot \sum_{s_2''} \Delta_a(s_1'', s_2'') \\ &= \tau(s_3, a)(s_3') \cdot \tau(s_1', a)(s_1'') \\ &= \tau(s_1' \parallel_\star s_3, a)(s_1'' \parallel_\star s_3'), \end{aligned}$$

and likewise we can show that

$$\sum_{s_1'' \parallel_\star s_3'} \Delta_a'(s_1'' \parallel_\star s_3', s_2'' \parallel_\star s_3'') = \tau(s_2' \parallel_\star s_3, a)(s_2'' \parallel_\star s_3'').$$

We have thus shown that R' is a ε-simulation relation, and hence

$$s_1 \parallel_\star s_3 \precsim_\varepsilon s_2 \parallel_\star s_3.$$

Clearly, this implies that $d(s_1 \parallel_\star s_3, s_2 \parallel_\star s_3) \leq \varepsilon$. ■

We conclude this section by exploring the computational aspects of composition of SMDPs. In particular, we would like to be able to also compute the distance between composite systems.

From Lemma D.5.1, we know that computing the simulation distance amounts to being able to compute the constants $c(F_s, F_{s'})$, for each pair of states s, s' of the SMDP. Hence we would like that, whenever two distributions μ and ν have effective CDFs then also their composition $\star(\mu, \nu)$ has an effective CDF. By Proposition D.3.5, it is easy to see that this holds for product, minimum, and maximum rate composition, since these compositions are still exponential distributions.

For maximum composition, the class $\mathcal{C}_\Lambda$ is unfortunately not closed under composition. However, the following result holds.

Proposition D.6.6. *Let $\star$ be maximum composition. For any $\mu, \nu, \eta \in \mathcal{C}_\Lambda$,*

1. *$c(F_\mu, F_{\star(\nu,\eta)})$ is computable and*
2. *$c(F_{\star(\mu,\eta)}, F_\nu)$ is computable.*

Proof.

1. This follows from Lemmas D.6.7, D.6.8, D.6.10, D.6.11, and D.6.12.
2. This follows from Lemmas D.6.7, D.6.8, D.6.13, D.6.14, and D.6.15. ■

The above results tells us that if we are interested in computing the distance $d(s_1, s_2 \parallel_\star s_3)$ or $d(s_1 \parallel_\star s_2, s_3)$, when $\star$ is maximum composition, then we can indeed compute the constants c that are needed for Algorithm D.5.1 to work.

We now state and prove the lemmas necessary to prove Proposition D.6.6.

Lemma D.6.7. *Let $\star$ be maximum composition, and let μ_1, μ_2, ν_1, and ν_2 be measures. The following holds for any $\varepsilon \in \mathbb{R}_{>0}$.*

1. *If one of μ_1 and μ_2 and one of ν_1 and ν_2 is the Dirac measure at 0, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$.*

2. *If one of μ_1 and μ_2 is the Dirac measure at 0, but none of ν_1 and ν_2 are, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$.*

3. *If none of μ_1 and μ_2 are the Dirac measure at 0, but one of ν_1 and ν_2 is, then $F_{\star(\mu_1,\mu_2)} \not\sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$.*

Proof. 1. We get $F_{\star(\mu_1,\mu_2)} = Dirac[0]$ and $F_{\star(\nu_1,\nu_2)} = Dirac[0]$, so we can use Proposition D.3.2.

2. We get $F_{\star(\mu_1,\mu_2)} = Dirac[0]$, so again we can use Proposition D.3.2.

3. We get $F_{\star(\nu_1,\nu_2)} = Dirac[0]$, and $F_{\star(\mu_1,\mu_2)} \neq Dirac[0]$, so once more we can use Proposition D.3.2. ■

Lemma D.6.8. *Let $\star$ be maximum composition, and let $F_{\mu_1} = Exp[\theta_1]$, $F_{\mu_2} = Exp[\theta_2]$, $F_{\nu_1} = Exp[\lambda_1]$, and $F_{\nu_2} = Exp[\lambda_2]$. Then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ where $\varepsilon = \frac{\max\{\lambda_1,\lambda_2\}}{\max\{\theta_1,\theta_2\}}$. Furthermore, this is the least ε such that the ε-faster-than relation holds.*

Proof. We have four cases to consider:

1. $\max\{\theta_1,\theta_2\} = \theta_1$ and $\max\{\lambda_1,\lambda_2\} = \lambda_1$, in which case $F_{\star(\mu_1,\mu_2)} = F_{\mu_1}$ and $F_{\star(\nu_1,\nu_2)} = F_{\nu_1}$.

2. $\max\{\theta_1,\theta_2\} = \theta_2$ and $\max\{\lambda_1,\lambda_2\} = \lambda_1$, in which case $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$ and $F_{\star(\nu_1,\nu_2)} = F_{\nu_1}$.

3. $\max\{\theta_1,\theta_2\} = \theta_1$ and $\max\{\lambda_1,\lambda_2\} = \lambda_2$, in which case $F_{\star(\mu_1,\mu_2)} = F_{\mu_1}$ and $F_{\star(\nu_1,\nu_2)} = F_{\nu_2}$.

4. $\max\{\theta_1,\theta_2\} = \theta_2$ and $\max\{\lambda_1,\lambda_2\} = \lambda_2$, in which case $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$ and $F_{\star(\nu_1,\nu_2)} = F_{\nu_2}$.

In all cases, the result then follows from Proposition D.3.5. ■

Lemma D.6.9. *Let $\star$ be maximum composition and let $F_\mu = Unif[a,b]$ and $F_\nu = Unif[c,d]$. If $a \leq c$ and $d \leq b$, then $F_{\star(\mu,\nu)}(t) \leq Unif[a,d](t)$ for all t.*

Proof. Note first that if $F_{\mu'} = Unif[a',b']$ and $F_{\nu'} = Unif[c',d']$ with $a' \leq c'$ and $b' \leq d'$, then clearly $Unif[a',b'](t) \geq Unif[c',d'](t)$ for all t.

Now, $a \leq a$ and $d \leq b$, so $Unif[a,d](t) \geq Unif[a,b](t)$ for all t. Likewise, $a \leq c$ and $d \leq d$, so $Unif[a,d] \geq Unif[c,d](t)$ for all t. Hence

$$Unif[a,d](t) \geq \max\{Unif[a,b](t), Unif[c,d](t)\} = F_{\star(\mu,\nu)}.$$ ■

Lemma D.6.10. *Let $\star$ be maximum composition, and let $F_\mu = Unif[a,b]$, $F_{\nu_1} = Unif[c_1,d_1]$, and $F_{\nu_2} = Unif[c_2,d_2]$.*

1. *If* $\min\{c_1, c_2\} = 0$ *and* $a > 0$, *then* $F_\mu \not\sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ *for any* ε.
2. *If* $\min\{c_1, c_2\} = 0$ *and* $a = 0$, *then* $F_\mu \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ *where* $\varepsilon = \frac{b}{\min\{d_1,d_2\}}$.
3. *If* $\min\{c_1, c_2\} > 0$, *then* $F_\mu \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ *where*

$$\varepsilon = \max\left\{\frac{a}{\min\{c_1, c_2\}}, \frac{b}{\min\{d_1, d_2\}}\right\}.$$

In all cases, this is the least ε *such that the* ε*-faster-than relation holds.*

Proof. 1. Take $t = \frac{a}{\varepsilon} > 0$ to get

$$F_\mu(\varepsilon \cdot t) = \mathit{Unif}[a, b](\varepsilon \cdot t) = \mathit{Unif}[a, b](a) = 0.$$

However, since $\min\{c_1, c_2\} = 0$, at least one of $\mathit{Unif}[c_1, d_1](t) > 0$ and $\mathit{Unif}[c_2, d_2](t) > 0$ must hold, and hence $F_{\star(\nu_1,\nu_2)}(t) > 0$.

2. We get

$$\begin{aligned} F_\mu(\varepsilon \cdot t) &= \mathit{Unif}\left[a \cdot \frac{\min\{d_1, d_2\}}{b}, \min\{d_1, d_2\}\right](t) \\ &= \mathit{Unif}[0, \min\{d_1, d_2\}](t) \\ &\geq F_{\star(\nu_1,\nu_2)}(t) && \text{by Lemma D.6.9.} \end{aligned}$$

If $\varepsilon' < \frac{b}{\min\{d_1,d_2\}}$, then

$$\begin{aligned} & F_\mu(\varepsilon' \cdot \min\{d_1, d_2\}) \\ &= \mathit{Unif}[a, b](\varepsilon' \cdot \min\{d_1, d_2\}) \\ &< \mathit{Unif}[a, b]\left(\frac{b}{\min\{d_1, d_2\}} \cdot \min\{d_1, d_2\}\right) \\ &= 1 \\ &= \max\{\mathit{Unif}[c_1, d_1](\min\{d_1, d_2\}), \mathit{Unif}[c_2, d_2](\min\{d_1, d_2\})\} \\ &= F_{\nu_1,\nu_2}(\min\{d_1, d_2\}). \end{aligned}$$

3. We consider each case separately.

Case $c_1 \leq c_2$ **and** $d_1 \leq d_2$**:** In this case we have $F_{\star(\nu_1,\nu_2)} = F_{\nu_1}$, so we can use Proposition D.3.5.

Case $c_1 \leq c_2$ **and** $d_1 > d_2$**:** In this case we get $\varepsilon = \max\left\{\frac{a}{c_1}, \frac{b}{d_2}\right\}$. If $\varepsilon = \frac{a}{c_1}$, then $\frac{c_1}{a} \leq \frac{d_2}{b}$, so

$$\begin{aligned} F_\mu(\varepsilon \cdot t) &= \mathit{Unif}\left[c_1, b \cdot \frac{d_2}{b}\right](t) \\ &\geq \mathit{Unif}\left[c_1, b \cdot \frac{d_2}{b}\right](t) \\ &= \mathit{Unif}[c_1, d_2](t) \\ &\geq F_{\star(\nu_1,\nu_2)}(t) && \text{by Lemma D.6.9.} \end{aligned}$$

For any $\varepsilon' < \frac{a}{c_1}$, let $t = \frac{a}{\varepsilon'} > c_1$. Then $F_\mu(\varepsilon' \cdot t) = F_\mu(a) = 0$, but $F_{\star(\nu_1,\nu_2)}(t) > 0$ since $c_1 \leq c_2$ and $t > c_1$.
On the other hand, if $\varepsilon = \frac{b}{d_2}$, then $\frac{d_2}{b} \leq \frac{c_1}{a}$. This means that

$$\begin{aligned} F_\mu(\varepsilon \cdot t) &= \mathit{Unif}\left[a \cdot \frac{d_2}{b}, d_2\right](t) \\ &\geq \mathit{Unif}\left[a \cdot \frac{c_1}{a}, d_2\right](t) \\ &= \mathit{Unif}[c_1, d_2](t) \\ &\geq F_{\star(\nu_1,\nu_2)}(t) && \text{by Lemma D.6.9.} \end{aligned}$$

For any $\varepsilon' < \frac{b}{d_2}$ we get

$$\begin{aligned} F_\mu(\varepsilon' \cdot d_2) &= \mathit{Unif}[a, b](\varepsilon' \cdot d_2) \\ &< \mathit{Unif}[a, b](\frac{b}{d_2} \cdot d_2) \\ &= 1 \\ &= F_{\star(\nu_1,\nu_2)}(d_2) \end{aligned}$$

because $d_1 > d_2$.

Case $c_1 > c_2$ **and** $d_1 \leq d_2$**:** In this case we get $\varepsilon = \max\left\{\frac{a}{c_2}, \frac{b}{d_1}\right\}$. If $\varepsilon = \frac{a}{c_2}$, then $\frac{c_2}{a} \leq \frac{d_1}{b}$, and hence

$$\begin{aligned} F_\mu(\varepsilon \cdot t) &= \mathit{Unif}\left[c_2, b \cdot \frac{c_2}{a}\right](t) \\ &\geq \mathit{Unif}\left[c_2, b \cdot \frac{d_1}{b}\right](t) \\ &= \mathit{Unif}[c_2, d_1](t) \\ &\geq F_{\star(\nu_1,\nu_2)}(t) && \text{by Lemma D.6.9.} \end{aligned}$$

For any $\varepsilon' < \frac{a}{c_2}$, let $t = \frac{a}{\varepsilon'} > c_2$ in order to get $F_\mu(\varepsilon' \cdot t) = \mathit{Unif}[a,b](a) = 0$, but $F_{\star(\nu_1,\nu_2)}(t) > 0$ since $c_1 > c_2$ and $t > c_2$.

On the other hand, if $\varepsilon = \frac{b}{d_1}$, then $\frac{d_1}{b} \leq \frac{c_2}{a}$. Then we get

$$\begin{aligned} F_\mu(\varepsilon \cdot t) &= \mathit{Unif}\left[a \cdot \frac{d_1}{b}, d_1\right](t) \\ &\geq \mathit{Unif}[c_2, d_1](t) \\ &\geq F_{\star(\nu_1,\nu_2)}(t) && \text{by Lemma D.6.9.} \end{aligned}$$

For any $\varepsilon' < \frac{b}{d_1}$ we get

$$\begin{aligned} F_\mu(\varepsilon' \cdot d_1) &= \mathit{Unif}[a,b](\varepsilon' \cdot d_1) \\ &< \mathit{Unif}[a,b](\frac{b}{d_1} \cdot d_1) \\ &= 1 \\ &= F_{\star(\nu_1,\nu_2)}(d_1) \end{aligned}$$

since $d_1 \leq d_2$.

Case $c_1 > c_2$ and $d_1 > d_2$: In this case we have $F_{\star(\nu_1,\nu_2)} = F_{\nu_2}$, so we can use Proposition D.3.5. ■

Lemma D.6.11. *Let $\star$ be maximum composition, and let $F_\mu = \mathit{Exp}[\theta]$, $F_{\nu_1} = \mathit{Unif}[a,b]$, $F_{\nu_2} = \mathit{Unif}[c,d]$. Then $F_\mu \not\sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ for any ε.*

Proof. $F_{\star(\nu_1,\nu_2)}(\min\{b,d\}) = 1$, but $F_\mu(t) < 1$ for all t, so

$$F_\mu(\varepsilon \cdot \min\{b,d\}) < F_{\star(\nu_1,\nu_2)}(\min\{b,d\})$$

for any ε. ■

Lemma D.6.12. *Let $\star$ be maximum composition, and let $\mu_1 = \mathit{Exp}[\theta_1]$, $\mu_2 = \mathit{Unif}[a,b]$, $\nu_1 = \mathit{Exp}[\theta_2]$, and $\nu_2 = \mathit{Unif}[c,d]$.*

1. *$F_{\mu_1} \not\sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ for any ε.*
2. *If $a > 0$, then $F_{\mu_2} \not\sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ for any ε.*
3. *If $a = 0$ and $\frac{1}{d-c} \geq \theta_2$, then $F_{\mu_2} \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ where $\varepsilon = \frac{b}{d}$.*
4. *If $a = 0$ and $\frac{1}{d-c} < \theta_2$, then $F_{\mu_2} \sqsubseteq_\varepsilon F_{\star(\nu_1,\nu_2)}$ where $\varepsilon = \theta_2 \cdot b$.*

In all cases, this is the least ε such that the ε-faster-than relation holds.

Proof. 1. $F_{\star(\nu_1,\nu_2)}(d) = 1$, but $F_{\mu_1}(t) < 1$ for all t, so $F_{\mu_1}(\varepsilon \cdot d) < F_{\star(\nu_1,\nu_2)}(d)$ for any ε.

2. If $a > 0$, let $\varepsilon \in \mathbb{R}_{>0}$ be given. Let $t = \frac{a}{\varepsilon} > 0$ to get $F_{\star(\nu_1,\nu_2)}(t) \geq Exp[\theta_2](t) > 0$ since $t > 0$, but $F_{\mu_2}(\varepsilon \cdot t) = Unif[a,b](a) = 0$, and hence $F_{\mu_2}(\varepsilon \cdot t) < F_{[\![\nu_1,\nu_2]\!]}(t)$.

3. If $a = 0$ and $\frac{1}{d-c} \geq \theta_2$, then the slope of $Unif[c,d]$ is greater than that of $Exp[\theta_2]$ until $Unif[c,d]$ hits 1 and flattens out. Hence $F_{\star(\nu_1,\nu_2)} = Unif[c,d]$, so we can use Proposition D.3.5 to get the result.

4. If $a = 0$ and $\frac{1}{d-c} < \theta_2$, let $\varepsilon = \theta_2 \cdot b$. By Proposition D.3.6 we then get $F_{\mu_2}(\varepsilon \cdot t) \geq Exp[\theta](t)$ for all t. Since the slope of $Unif\left[\frac{a}{\varepsilon}, \frac{b}{\varepsilon}\right]$ is θ_2, it has greater slope than $Unif[c,d]$, and hence also $F_{\mu_2}(\varepsilon \cdot t) = Unif\left[\frac{a}{\varepsilon}, \frac{b}{\varepsilon}\right](t) \geq Unif[c,d](t)$. We therefore get

$$F_{\mu_2}(\varepsilon \cdot t) \geq \max\{Exp[\theta_2](t), Unif[c,d](t)\} = F_{\star(\nu_1,\nu_2)}(t).$$

If $\varepsilon' < \theta_2 \cdot b$, then the slope in 0 of $F_{\mu_2}(\varepsilon' \cdot t) = Unif\left[\frac{a}{\varepsilon'}, \frac{b}{\varepsilon'}\right](t)$ must be less than that of $Exp[\theta_2]$. Hence there exists some $t > 0$ sufficiently close to 0 such that $F_{\mu_2}(\varepsilon' \cdot t) < Exp[\theta_2](t) = F_{\star(\nu_1,\nu_2)}(t)$. ■

Lemma D.6.13. *Let $\star$ be maximum composition, and let $F_{\mu_1} = Unif[a_1,b_1]$, $F_{\mu_2} = Unif[a_2,b_2]$, and $F_\nu = Unif[c,d]$.*

1. *If $c = 0$ and $\min\{a_1,a_2\} > 0$, then $F_{\star(\mu_1,\mu_2)} \not\sqsubseteq_\varepsilon F_\nu$ for any ε.*
2. *If $c = 0$ and $a_1 = a_2 = 0$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \frac{\min\{b_1,b_2\}}{d}$.*
3. *If $c = 0$ and $\min\{a_1,a_2\} = 0$ and either $a_1 < a_2$ and $b_1 \leq b_2$ or $a_1 > a_2$ and $b_1 > b_2$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \frac{\min\{b_1,b_2\}}{d}$.*
4. *If $c = 0$ and $\min\{a_1,a_2\} = 0$ and either $a_1 < a_2$ and $b_1 > b_2$ or $a_1 > a_2$ and $b_1 \leq b_2$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \frac{\max\{b_1,b_2\}}{d}$.*
5. *If $c > 0$, $a_1 < a_2$, $b_1 > b_2$, $\frac{1}{c-\frac{a_1 \cdot c}{b_1}} \geq \frac{1}{d-c}$, and $\frac{1}{d-\frac{a_2 \cdot d}{b_2}} \leq \frac{1}{d-c}$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \min\left\{\frac{a_1}{c}, \frac{b_2}{d}\right\}$.*
6. *If $c > 0$, $a_1 < a_2$, $b_1 > b_2$, $\frac{1}{c-\frac{a_1 \cdot c}{b_1}} < \frac{1}{d-c}$, and $\frac{1}{d-\frac{a_2 \cdot d}{b_2}} \leq \frac{1}{d-c}$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \frac{b_2}{d}$.*
7. *If $c > 0$, $a_1 < a_2$, $b_1 > b_2$, $\frac{1}{c-\frac{a_1 \cdot c}{b_1}} \geq \frac{1}{d-c}$, and $\frac{1}{d-\frac{a_2 \cdot d}{b_2}} > \frac{1}{d-c}$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ where $\varepsilon = \frac{a_1}{c}$.*

8. *If* $c > 0$, $a_1 < a_2$, $b_1 > b_2$, $\frac{1}{c-\frac{a_1 \cdot c}{b_1}} < \frac{1}{d-c}$, *and* $\frac{1}{d-\frac{a_2 \cdot d}{b_2}} > \frac{1}{d-c}$, *then* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where*

$$\varepsilon = \frac{(b_1 - a_1) \cdot k}{d \cdot k - c \cdot k - d \cdot a_1 + b_1 \cdot c}$$

and $k = \frac{a_1 \cdot b_2 - a_2 \cdot b_1}{a_1 - b_1 - a_2 + b_2}$.

9. *If* $c > 0$, $a_1 > a_2$, $b_1 < b_2$, $\frac{1}{c-\frac{a_2 \cdot c}{b_2}} \geq \frac{1}{d-c}$, *and* $\frac{1}{d-\frac{a_1 \cdot d}{b_1}} \leq \frac{1}{d-c}$, *then* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where* $\varepsilon = \min\left\{\frac{a_2}{c}, \frac{b_1}{d}\right\}$.

10. *If* $c > 0$, $a_1 > a_2$, $b_1 < b_2$, $\frac{1}{c-\frac{a_2 \cdot c}{b_2}} < \frac{1}{d-c}$, *and* $\frac{1}{d-\frac{a_1 \cdot d}{b_1}} \leq \frac{1}{d-c}$, *then* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where* $\varepsilon = \frac{b_1}{d}$.

11. *If* $c > 0$, $a_1 > a_2$, $b_1 < b_2$, $\frac{1}{c-\frac{a_2 \cdot c}{b_2}} \geq \frac{1}{d-c}$, *and* $\frac{1}{d-\frac{a_1 \cdot d}{b_1}} > \frac{1}{d-c}$, *then* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where* $\varepsilon = \frac{a_2}{c}$.

12. *If* $c > 0$, $a_1 > a_2$, $b_1 < b_2$, $\frac{1}{c-\frac{a_2 \cdot c}{b_2}} < \frac{1}{d-c}$, *and* $\frac{1}{d-\frac{a_1 \cdot d}{b_1}} > \frac{1}{d-c}$, *then* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where*

$$\varepsilon = \frac{(b_2 - a_2) \cdot k}{d \cdot k - c \cdot k - d \cdot a_2 + b_2 \cdot c}$$

and $k = \frac{a_1 \cdot b_2 - a_2 \cdot b_1}{a_1 - b_1 - a_2 + b_2}$.

13. *Otherwise,* $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_\nu$ *where*

$$\varepsilon = \max\left\{\frac{\min\{a_1, a_2\}}{c}, \frac{\min\{b_1, b_2\}}{d}\right\}.$$

In all cases, this is the least ε *such that the* ε*-faster-than relation holds.*

Proof. 1. Take an arbitrary $\varepsilon \in \mathbb{R}_{>0}$ and let $t = \frac{\min\{a_1,a_2\}}{\varepsilon} > 0$. Then $F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) = F_{\star(\mu_1,\mu_2)}(\min\{a_1, a_2\}) = 0$, but $F_\nu(t) > 0$ since $c = 0$ and $t > 0$. Hence $F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) < F_\nu(t)$.

2. If $a_1 = a_2 = 0$, then $F_{\star(\mu_1,\mu_2)} = \mathit{Unif}[0, \min\{b_1, b_2\}]$, so the result follows from Proposition D.3.5.

3. If $a_1 \leq a_2$ and $b_1 \leq b_2$, then $F_{\star(\mu_1,\mu_2)} = F_{\mu_1}$, and if $a_1 > a_2$ and $b_1 > b_2$, then $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$. In either case, we can then use Proposition D.3.5 to obtain the result.

4. We consider here the case where $a_1 < a_2$ and $b_1 > b_2$. The case where $a_1 > a_2$ and $b_1 \leq b_2$ is symmetrical, noting that if $b_1 = b_2$, then $F_{\star(\mu_1,\mu_2)} = F_{\mu_1}$, in which case the result follows from Proposition D.3.5. We have $c = \min\{a_1, a_2\} = a_1 = 0$ and $\varepsilon = \frac{b_1}{d}$. Then

$$\begin{aligned} F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) &\geq \mathit{Unif}[a_1, b_1]\,(\varepsilon \cdot t) \\ &= \mathit{Unif}\left[a_1 \cdot \frac{d}{b_1}, b_1 \cdot \frac{d}{b_1}\right](t) \\ &= \mathit{Unif}[0, d]\,(t) \\ &= \mathit{Unif}[c, d]\,(t). \end{aligned}$$

To see that this is the least ε such that the ε-faster-than relation holds, first note that $\mathit{Unif}\left[\frac{a_1}{\varepsilon}, \frac{b_1}{\varepsilon}\right]$ and $\mathit{Unif}\left[\frac{a_2}{\varepsilon}, \frac{b_2}{\varepsilon}\right]$ cross in the point

$$t = \frac{a_1 \cdot b_2 - a_2 \cdot b_1}{\varepsilon \cdot (a_1 - b_1 - a_2 + b_2)}$$

with $0 = a_1 < a_2 < t < b_2 < b_1$. From this it follows that

$$1 > \mathit{Unif}[c, d]\,(t) = \mathit{Unif}[a_1, b_2]\,(\varepsilon \cdot t) = \mathit{Unif}[a_2, b_2]\,(\varepsilon \cdot t) > 0.$$

Hence, if $\varepsilon' < \varepsilon$ we get

$$\mathit{Unif}[c, d]\,(t) = \mathit{Unif}[a_1, b_1]\,(\varepsilon \cdot t) > \mathit{Unif}[a_1, b_2]\,(\varepsilon' \cdot t)$$

and

$$\mathit{Unif}[c, d]\,(t) = \mathit{Unif}[a_2, b_2]\,(\varepsilon \cdot t) > \mathit{Unif}[a_2, b_2]\,(\varepsilon' \cdot t),$$

and therefore $F_{\star(\mu_1,\mu_2)}(\varepsilon' \cdot t) < \mathit{Unif}[c, d]\,(t)$.

5. $\frac{1}{c - \frac{a_1 \cdot c}{b_1}} \geq \frac{1}{d-c}$ means that

$$\mathit{Unif}[a_1, b_2]\left(\frac{a_1}{c} \cdot t\right) = \mathit{Unif}\left[c, b_1 \cdot \frac{c}{a_1}\right](t)$$

has greater slope than $\mathit{Unif}[c, d]\,(t)$, so

$$\mathit{Unif}[a_1, b_1]\left(\frac{a_1}{c} \cdot t\right) \geq \mathit{Unif}[c, d]\,(t).$$

Likewise, $\frac{1}{d - \frac{a_2 \cdot d}{b_2}} \leq \frac{1}{d-c}$ means that

$$\mathit{Unif}[a_2, b_2]\left(\frac{b_2}{d} \cdot t\right) = \mathit{Unif}\left[a_2 \cdot \frac{d}{b_2}, d\right](t)$$

has smaller slope than $Unif[c,d](t)$, and hence

$$Unif[a_2,b_2]\left(\frac{b_2}{d}\cdot t\right) \geq Unif[c,d](t).$$

We therefore conclude

$$F_{\star(\mu_1,\mu_2)}\left(\min\left\{\frac{a_1}{c},\frac{b_2}{d}\right\}\cdot t\right) \geq Unif[c,d](t).$$

If $\varepsilon' < \varepsilon$, first assume that $\varepsilon = \frac{a_1}{c}$ and let $t = \frac{a_1}{\varepsilon'} > c$. Then

$$F_{\star(\mu_1,\mu_2)}(\varepsilon'\cdot t) = F_{\star(\mu_1,\mu_2)}(a_1) = 0,$$

but $Unif[c,d](t) > 0$ since $t > c$. Now assume that $\varepsilon = \frac{b_2}{d}$. Then we get

$$F_{\star(\mu_1,\mu_2)}(\varepsilon'\cdot d) < F_{\star(\mu_1,\mu_2))}(\varepsilon\cdot d) = F_{\star(\mu_1,\mu_2)}(b_2) = 1 = Unif[c,d](d).$$

6. This case is the same as case 5, only considering the part where $\varepsilon = \frac{b_2}{d}$.
7. This case is the same as case 5, only considering the part where $\varepsilon = \frac{a_1}{c}$.
8. ε and k are chosen such that $\frac{k}{\varepsilon}$ is the point in which $Unif\left[\frac{a_1}{\varepsilon},\frac{b_1}{\varepsilon}\right]$, $Unif\left[\frac{a_2}{\varepsilon},\frac{b_2}{\varepsilon}\right]$, and $Unif[c,d]$ cross. Hence we get

$$F_{\star(\mu_1,\mu_2)}\left(\varepsilon\cdot\frac{k}{\varepsilon}\right) = Unif[c,d]\left(\frac{k}{\varepsilon}\right).$$

We have

$$F_{\star(\mu_1,\mu_2)}(\varepsilon\cdot t) = Unif[a_1,b_1](\varepsilon\cdot t) \geq Unif[c,d](t)$$

for any $t \geq \frac{k}{\varepsilon}$ and

$$F_{\star(\mu_1,\mu_2)}(\varepsilon\cdot t) = Unif[a_2,b_2](\varepsilon\cdot t) \geq Unif[c,d](t)$$

for any $t \leq \frac{k}{\varepsilon}$. Hence we can conclude that $F_{\star(\mu_1,\mu_2)}(\varepsilon\cdot t) \geq Unif[c,d](t)$. If $\varepsilon' < \varepsilon$, then $F_{\star(\mu_1,\mu_2)}\left(\varepsilon'\cdot\frac{k}{\varepsilon}\right) < F_{\star(\mu_1,\mu_2)}\left(\varepsilon\cdot\frac{k}{\varepsilon}\right) = Unif[c,d]\left(\frac{k}{\varepsilon}\right)$.

9. Symmetric to case 5.
10. Symmetric to case 6.
11. Symmetric to case 7.
12. Symmetric to case 8.

13. In this case, we either get $F_{\star(\mu_1,\mu_2)} = F_{\mu_1}$ or $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$, and the result is then obtained by applying Proposition D.3.5. ■

Lemma D.6.14. *Let $\star$ be maximum composition, and let $F_\mu = Exp[\theta]$, $F_{\nu_1} = Unif[a,b]$, $F_{\nu_2} = Unif[c,d]$.*

1. *If $\min\{a,c\} > 0$, then $F_{\star(\nu_1,\nu_2)} \not\sqsubseteq_\varepsilon F_\mu$ for any ε.*
2. *If $a = 0$ and $c > 0$, then $F_{\star(\nu_1,\nu_2)} \sqsubseteq_\varepsilon F_\mu$ where $\varepsilon = \theta \cdot b$.*
3. *If $a > 0$ and $c = 0$, then $F_{\star(\nu_1,\nu_2)} \sqsubseteq_\varepsilon F_\mu$ where $\varepsilon = \theta \cdot d$.*
4. *If $a = 0$ and $c = 0$, then $F_{\star(\nu_1,\nu_2)} \sqsubseteq_\varepsilon F_\mu$ where $\varepsilon = \theta \cdot \min\{b,d\}$.*

In all cases, this is the least ε such that the ε-faster-than relation holds.

Proof. 1. If $\min\{a,c\} > 0$, let ε be given, and let $t = \frac{\min\{a,c\}}{\varepsilon} > 0$. Then

$$F_{\star(\nu_1,\nu_2)}(\varepsilon \cdot t) = F_{\star(\nu_1,\nu_2)}(\min\{a,c\}) = 0$$

but $F_\mu(t) = Exp[\theta]\,(t) > 0$ since $t > 0$.

2. By Proposition D.3.6, we know that $Unif[a,b] \sqsubseteq_\varepsilon Exp[\theta]$. Hence

$$F_{\star(\nu_1,\nu_2)}(\varepsilon \cdot t) \geq Unif[a,b]\,(\varepsilon \cdot t) \geq Exp[\theta]\,(t).$$

If $\varepsilon' < \theta \cdot b$, then there must exist some $t > 0$ sufficiently close to 0 such that

$$F_{\star(\nu_1,\nu_2)}(\varepsilon' \cdot t) = Unif[a,b]\,(\varepsilon' \cdot t) < Exp[\theta]\,(t).$$

3. Similar to the case where $a = 0$ and $c > 0$.
4. If $a = 0$ and $c = 0$, then we get $F_{\star(\nu_1,\nu_2)} = Unif[a,b]$ if $b \leq d$ and $F_{\star(\nu_1,\nu_2)} = Unif[c,d]$ if $b > d$. In either case, we can use Proposition D.3.6 to obtain the desired result. ■

Lemma D.6.15. *Let $\star$ be maximum composition, and let $\mu_1 = Exp[\theta_1]$, $\mu_2 = Unif[a,b]$, $\nu_1 = Exp[\theta_2]$, and $\nu_2 = Unif[c,d]$.*

1. *If $a = 0$ and $\frac{1}{b-a} \geq \theta_1$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\nu_1}$ where $\varepsilon = \theta_2 \cdot b$.*
2. *If $a > 0$ or $\frac{1}{b-a} < \theta_1$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\nu_1}$ where $\varepsilon = \frac{\theta_2}{\theta_1}$.*
3. *If $\frac{1}{b-a} \geq \theta_1$ and $a = 0$, then $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\nu_2}$ where $\varepsilon = \frac{b}{d}$.*
4. *Otherwise, $F_{\star(\mu_1,\mu_2)} \sqsubseteq_\varepsilon F_{\nu_2}$ where*

$$\varepsilon = \max\left\{\frac{b}{d}, \frac{(b-a)\cdot k}{d\cdot k - c\cdot k - d\cdot a + b\cdot c}\right\},$$

$k = \frac{b\cdot\theta_1 + W(\theta_1\cdot(a-b)\cdot e^{-b\cdot\theta_1})}{\theta_1}$, and W is the Lambert W-function.

In all cases, this is the least ε such that the ε-faster-than relation holds.

Proof. 1. In this case, $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$, so we can use Proposition D.3.6 to obtain the result.

2. We get

$$F_{\star(\mu_1,\mu_2)}\left(\frac{\theta_2}{\theta_1}\cdot t\right) \geq Exp[\theta_1]\left(\frac{\theta_2}{\theta_1}\cdot t\right) = Exp[\theta_2](t) = F_{\nu_1}(t).$$

To see that this is the least ε such that the ε-faster-than relation holds, first note that because $a > 0$ or $\frac{1}{b-a} < \theta_1$, there must be some interval $[0,t]$ where $F_{\star(\mu_1,\mu_2)}(t') = Exp[\theta_1](t')$ for all $t' \in [0,t]$. If $\varepsilon' < \varepsilon$, then let $t' \in \left[0, \frac{t}{\varepsilon}\right]$, so that $\varepsilon \cdot t' \in [0,t]$, and also $\varepsilon' \cdot t' \in [0,t]$. Then we get

$$F_{\star(\mu_1,\mu_2)}(\varepsilon' \cdot t') = Exp[\theta_1](\varepsilon' \cdot t') < Exp[\theta_1](\varepsilon \cdot t') = Exp[\theta_2](t').$$

3. We get $F_{\star(\mu_1,\mu_2)} = F_{\mu_2}$, so we can use Proposition D.3.5.

4. In this case, F_{μ_1} and F_{μ_2} will cross in some non-zero point. k is chosen so that

$$F_{\mu_1}(\varepsilon^* \cdot t^*) = F_{\mu_2}(\varepsilon^* \cdot t^*) = F_{\nu_2}(t^*)$$

where

$$\varepsilon^* = \frac{(b-a)\cdot k}{d\cdot k - c\cdot k - d\cdot a + b\cdot c}$$

and

$$t^* = \frac{k}{\varepsilon^*}.$$

This also means that

$$F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) = \begin{cases} F_{\mu_1}(\varepsilon \cdot t) & \text{if } t \leq t^* \\ F_{\mu_2}(\varepsilon \cdot t) & \text{if } t \geq t^*. \end{cases}$$

Now, if $\varepsilon = \frac{b}{d}$, then $F_{\mu_2}(\varepsilon \cdot d) = Unif\left[a \cdot \frac{d}{b}, d\right](d) = Unif[c,d](d)$, and $F_{\mu_2}(\varepsilon \cdot t^*) \geq F_{\mu_2}(\varepsilon^* \cdot t^*) = F_{\nu_2}(t^*)$. Hence $F_{\mu_2}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$ for all $t \geq t^*$. For $t \leq t^*$ we get $F_{\mu_1}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$, and hence $F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$. If $\varepsilon' < \varepsilon$, then $F_{\star(\mu_1,\mu_2)}(\varepsilon' \cdot d) < F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot d) = 1 = Unif[c,d](d)$.
If $\varepsilon = \varepsilon^*$, then $F_{\mu_1}(\varepsilon \cdot t^*) = F_{\mu_2}(\varepsilon \cdot t^*) = F_{\nu_2}(t^*)$. Since

$$F_{\mu_2}(\varepsilon \cdot d) \geq F_{\mu_2}\left(\frac{b}{d}\cdot d\right) = 1 = Unif[c,d](d),$$

we get $F_{\mu_2}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$ for all $t \geq t^*$. For $t \leq t^*$ we get $F_{\mu_1}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$, and hence we conclude $F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t) \geq F_{\nu_2}(t)$. If $\varepsilon' < \varepsilon$, then $F_{\star(\mu_1,\mu_2)}(\varepsilon' \cdot t^*) < F_{\star(\mu_1,\mu_2)}(\varepsilon \cdot t^*) = F_{\nu_2}(t^*)$. ■

D.7 Logical Properties of the Simulation Distance

If the distance between two processes is small, then we would also expect that they satisfy almost the same properties. In order to make this idea precise, in this section we introduce and study a slight extension of Markovian logic [17], which we will call *timed Markovian logic* (TML). The syntax of TML is given by the following grammar, where $\alpha \in \mathcal{AP}$, $p \in \mathbb{Q}_{\geq 0} \cap [0,1]$, $t \in \mathbb{Q}_{\geq 0}$, and $a \in A$.

$$\text{TML}: \quad \varphi ::= \alpha \mid \neg\alpha \mid \ell_p t \mid m_p t \mid L^a_p \varphi \mid M^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

The semantics of TML is given by

$$\begin{array}{llll} s \models \alpha & \text{iff} & \alpha \in L(s) & s \models \ell_p t \quad \text{iff} \quad F_s(t) \geq p \\ s \models \neg\alpha & \text{iff} & \alpha \notin L(s) & s \models m_p t \quad \text{iff} \quad F_s(t) \leq p \\ s \models \varphi \wedge \varphi' & \text{iff} & s \models \varphi \text{ and } s \models \varphi' & s \models L^a_p \varphi \quad \text{iff} \quad \tau(s,a)([\![\varphi]\!]) \geq p \\ s \models \varphi \vee \varphi' & \text{iff} & s \models \varphi \text{ or } s \models \varphi' & s \models M^a_p \varphi \quad \text{iff} \quad \tau(s,a)([\![\varphi]\!]) \leq p \end{array}$$

where $[\![\varphi]\!] = \{s \in S \mid s \models \varphi\}$ is the set of states satisfying φ.

We also isolate the following two fragments of TML.

$$\text{TML}^{\geq}: \quad \varphi ::= \alpha \mid \neg\alpha \mid \ell_p t \mid L^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

$$\text{TML}^{\leq}: \quad \varphi ::= \alpha \mid \neg\alpha \mid m_p t \mid M^a_p \varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi'$$

Intuitively, the modal formula $L^a_p \varphi$ says that the probability of taking an a-transition to where φ holds is *at least* p, and $M^a_p \varphi$ says the probability is *at most* p. $\ell_p t$ and $m_p t$ are similar in spirit, but talk about the probability of firing a transition instead. Thus, $\ell_p t$ says that the probability of firing a transition before time t is *at least* p, whereas $m_p t$ says that the probability is *at most* p.

For any $\varphi \in \text{TML}$ and $\varepsilon \geq 1$ we denote the *ε-perturbation* of φ by $(\varphi)_\varepsilon$ and define it inductively as

$$\begin{array}{ll} (\alpha)_\varepsilon = \alpha & (\ell_p t)_\varepsilon = \ell_p \varepsilon \cdot t \\ (\neg\alpha)_\varepsilon = \neg\alpha & (m_p t)_\varepsilon = m_p \varepsilon \cdot t \\ (\varphi \wedge \varphi')_\varepsilon = (\varphi)_\varepsilon \wedge (\varphi')_\varepsilon & (L^a_p \varphi)_\varepsilon = L^a_p (\varphi)_\varepsilon \\ (\varphi \vee \varphi')_\varepsilon = (\varphi)_\varepsilon \vee (\varphi')_\varepsilon & (M^a_p \varphi)_\varepsilon = M^a_p (\varphi)_\varepsilon. \end{array}$$

By making use of the alternative characterisation for simulation given in Proposition D.2.3 and drawing upon ideas from [7], we can now prove the following logical characterisation of the ε-simulation relation.

Theorem D.7.1. *Let $\varepsilon \in \mathbb{Q}_{\geq 0}$ with $\varepsilon \geq 1$. Then the following holds.*

- $s_1 \precsim_\varepsilon s_2$ *if and only if* $\forall \varphi \in \text{TML}^{\geq}. s_1 \models \varphi \implies s_2 \models (\varphi)_\varepsilon$.

- $s_1 \precsim_\varepsilon s_2$ *if and only if* $\forall \varphi \in \mathrm{TML}^{\leq}.s_2 \models (\varphi)_\varepsilon \implies s_1 \models \varphi$.

Proof. • We first prove the first item.

($\implies$) We proceed by induction on φ. The cases of conjunction and disjunction are standard.

Case $\varphi = \alpha$: $s_1 \models \alpha$ means that $\alpha \in L(s_1)$. Since $L(s_1) = L(s_2)$ we then get $s_2 \models \alpha$.

Case $\varphi = \neg\alpha$: $s_1 \models \neg\alpha$ means that $\alpha \notin L(s_1)$. Since $L(s_1) = L(s_2)$ we then get $s_2 \models \neg\alpha$.

Case $\varphi = \ell_p t$: $s_1 \models \ell_p t$ means that $F_{s_1}(t) \geq p$, and since $F_{s_2}(\varepsilon \cdot t) \geq F_{s_1}(t)$, we get $s_2 \models (\varphi)_\varepsilon$.

Case $\varphi = L^a_p \varphi'$: $s_1 \models L^a_p \varphi'$ means that $\tau(s_1, a)([\![\varphi']\!]) \geq p$. There exists a coupling $\Delta_a(s, s')$ such that

$$\begin{aligned}
\tau(s_1, a)([\![\varphi']\!]) &= \sum_{s \in [\![\varphi']\!]} \tau(s_1, a)(s) \\
&= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in S} \Delta_a(s, s') \\
&= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in [\![(\varphi')_\varepsilon]\!]} \Delta_a(s, s') && \text{(ind. hyp.)} \\
&\leq \sum_{s' \in [\![(\varphi')_\varepsilon]\!]} \tau(s_2, a)(s') \\
&= \tau(s_2, a)([\![(\varphi')_\varepsilon]\!]),
\end{aligned}$$

and hence $s_2 \models (\varphi)_\varepsilon$.

($\impliedby$) We construct the relation

$$\mathcal{R} = \{(s, s') \in S \times S \mid \forall \varphi \in \mathrm{TML}^{\geq}.s \models \varphi \implies s' \models \varphi\}$$

and we must show that it is a ε-simulation relation. Let $(s_1', s_2') \in \mathcal{R}$ be arbitrary. We will first show that $L(s_1') = L(s_2')$. If $\alpha \in L(s_1')$, then $s_1' \models \alpha$, and hence $s_2' \models \alpha$, which means that $\alpha \in L(s_2')$. If $\alpha \notin L(s_1')$, then $s_1' \models \neg\alpha$, implying that $s_2' \models \neg\alpha$, so $\alpha \notin L(s_2')$. Therefore $L(s_1') = L(s_2')$.

Next we will show that $F_{s_1'}(t) \leq F_{s_2'}(\varepsilon \cdot t)$ for all $t \in \mathbb{R}_{\geq 0}$. Assume towards a contradiction that $F_{s_1'}(t) > F_{s_2'}(\varepsilon \cdot t)$ for some $t \in \mathbb{Q}_{\geq 0}$. Then there exists $q \in \mathbb{Q}_{\geq 0}$ such that $F_{s_1'}(t) > q > F_{s_2'}(\varepsilon \cdot t)$. But then $s_1' \models \ell_q t$ whereas $s_2' \not\models \ell_q \varepsilon \cdot t$, which contradicts how $\mathcal{R}$ was constructed. Hence $F_{s_1'}(t) \leq F_{s_2'}(\varepsilon \cdot t)$ for all $t \in \mathbb{Q}_{\geq 0}$. Now assume towards a contradiction that $F_{s_1'}(t) > F_{s_2'}(\varepsilon \cdot t)$ for some $t \in \mathbb{R}_{\geq 0}$ and let $\varepsilon' = F_{s_1'}(t) - F_{s_2'}(\varepsilon \cdot t) > 0$. By right-continuity, there exists $\delta > 0$ such that $\varepsilon \cdot t < c < \varepsilon \cdot t + \delta$ implies $|F_{s_2'}(c) - F_{s_2'}(\varepsilon \cdot t)| < \varepsilon'$. Now pick some $q \in \mathbb{Q}_{\geq 0}$ such that $\varepsilon \cdot t < q < \varepsilon \cdot t + \delta$. Then

$$F_{s_2'}(\varepsilon \cdot t) \leq F_{s_2'}(q) < F_{s_1'}(t) \leq F_{s_1'}(q),$$

which is a contradiction. Hence we conclude that $F_{s_1'}(t) \leq F_{s_2'}(\varepsilon \cdot t)$ for all $t \in \mathbb{R}_{\geq 0}$.

Finally we will show that $\tau(s_1', a)(C) \leq \tau(s_2', a)(R(C))$ for all a and $C \subseteq S$. Pick an arbitrary a and $C \subseteq S$. By construction of $\mathcal{R}$, we know that

$$\tau(s_1', a)([\![\varphi]\!]) \geq p \quad \text{implies} \quad \tau(s_2', a)([\![(\varphi)_\varepsilon]\!]) \geq p \quad \text{for any } p.$$

This implies that

$$\tau(s_1', a)([\![\varphi]\!]) \leq \tau(s_2', a)([\![(\varphi)_\varepsilon]\!]) \tag{D.2}$$

for all $\varphi \in \mathrm{TML}^{\geq}$. The strategy will now be to construct a formula φ to exploit the inequality in Equation (D.2). To do this, we introduce the following notation. For a state $s \in S$, let

$$(\!|s|\!) = \{\varphi \in \mathrm{TML}^{\geq} \mid s \models \varphi\} \text{ and } (\!|s|\!)^{\varepsilon} = \{\varphi \in \mathrm{TML}^{\geq} \mid s \models (\varphi)_\varepsilon\}.$$

Given a formula $\varphi \in \mathrm{TML}^{\geq}$, we let $\mathbb{Q}(\varphi)$ be the set of values $t \in \mathbb{Q}_{\geq 0}$ and $p \in \mathbb{Q}_{\geq 0} \cap [0,1]$ that are used in φ. Furthermore, we define the depth $dpt(\varphi)$ as

$$dpt(\varphi) = \begin{cases} 0 & \text{if } \varphi = \ell_p t \text{ or } \varphi = m_p t, \\ 1 + dpt(\varphi') & \text{if } \varphi = L_p^a \varphi',\ \varphi = M_p^a \varphi', \\ & \quad \text{or } \varphi = \neg \varphi', \\ 1 + \max\{dpt(\varphi_1), dpt(\varphi_2)\} & \text{if } \varphi = \varphi_1 \wedge \varphi_2. \end{cases}$$

Finally, we let

$$I_k = \{q \in \mathbb{Q}_{\geq 0} \mid q = l \cdot \frac{1}{j} \text{ for some } l \in \mathbb{N}_0 \text{ and } j \in \mathbb{N} \text{ where } l \leq k \text{ and } j \leq k\}.$$

Then we can define

$$F_k = \{\varphi \in \mathrm{TML}^{\geq} \mid dpt(\varphi) \leq k \text{ and } \mathbb{Q} \subseteq I_k\}$$

as a finite fragment of $\mathrm{TML}^{\geq}$ and

$$(\!|s|\!)_k = (\!|s|\!) \cap F_k = \{\varphi \in F_k \mid s \models \varphi\}$$

as the restriction of $(\!|s|\!)$ to F_k. Intuitively, F_k is a better and better approximation of all formulas of $\mathrm{TML}^{\geq}$ as k increases. Formally, this means that $\bigcup_{k \in \mathbb{N}} F_k = \mathrm{TML}^{\geq}$, and hence any formula in $\mathrm{TML}^{\geq}$ will be in one of the F_k for some k. Now we can construct a formula that describes the set $R(C)$. Note that by construction of $\mathcal{R}$ we have

$$\mathcal{R}(C) = \{s' \in S \mid \exists s \in C.(s, s') \in \mathcal{R}\} = \{s' \in S \mid \exists s \in C.(\!|s|\!) \subseteq (\!|s'|\!)^{\varepsilon}\}$$

and we also have

$$\bigcup_{s\in C}\bigcap_{\varphi\in(\!|s|\!)}[\![(\varphi)_\varepsilon]\!] = \bigcup_{s\in C}\{s' \in S \mid (\!|s|\!) \subseteq (\!|s'|\!)^\varepsilon\} = \{s' \in S \mid \exists s \in C.(\!|s|\!) \subseteq (\!|s'|\!)^\varepsilon\},$$

so $\mathcal{R}(C) = \bigcup_{s\in C}\bigcap_{\varphi\in(\!|s|\!)}[\![(\varphi)_\varepsilon]\!]$.

We consider the case where C is finite and the case where C is infinite separately. Assume first that C is finite. Now let

$$\chi^C_k = \bigvee_{s\in C}\bigwedge_{\varphi\in(\!|s|\!)_k}\varphi, \quad (\chi^C_k)_\varepsilon = \bigvee_{s\in C}\bigwedge_{\varphi\in(\!|s|\!)_k}(\varphi)_\varepsilon, \quad \text{and}$$

$$\chi^C = \bigvee_{s\in C}\bigwedge_{\varphi\in(\!|s|\!)}\varphi.$$

Because C is finite, χ^C_k, $(\chi^C_k)_\varepsilon$, and χ^C consist of finitely many disjunctions and conjunctions and are hence formulas. These formulas will be the ones we use in Equation (D.2). Note that $[\![\chi^C]\!] \subseteq [\![\chi^C_k]\!]$ for any k. Now let

$$C_k = [\![\chi^C_k]\!] \quad \text{and} \quad C^\varepsilon_k = [\![(\chi^C_k)_\varepsilon]\!].$$

Then we get decreasing chains

$$C_1 \supseteq C_2 \supseteq \ldots \quad \text{and} \quad C^\varepsilon_1 \supseteq C^\varepsilon_2 \supseteq \ldots$$

of finite sets, and we will now prove that $\bigcap_{k\in\mathbb{N}} C^\varepsilon_k = \mathcal{R}(C)$. If $s' \in \mathcal{R}(C)$, then there exists $s \in C$ such that $(\!|s|\!) \subseteq (\!|s'|\!)^\varepsilon$, and hence $s' \models \bigwedge_{\varphi\in(\!|s|\!)_k}(\varphi)_\varepsilon$ for all k, so $s' \in \bigcap_{k\in\mathbb{N}} C^\varepsilon_k$. If $s' \notin \mathcal{R}(C)$, then for all $s \in C$ there exists $\varphi_s \in \mathrm{TML}^{\geq}$ such that $s \models \varphi_s$ but $s' \not\models (\varphi_s)_\varepsilon$. Since C is finite, we can fix k' such that $\varphi_s \in F_{k'}$ for all $s \in C$. Then $s' \notin C^\varepsilon_{k'}$ because $s' \not\models \bigvee_{s\in C}\bigwedge_{\varphi\in(\!|s|\!)_{k'}}(\varphi)_\varepsilon$ since $\varphi_s \in (\!|s|\!)_{k'}$. Therefore $s' \notin \bigcap_{k\in\mathbb{N}} C^\varepsilon_k$, so we conclude that $\mathcal{R}(C) = \bigcap_{k\in\mathbb{N}} C^\varepsilon_k$.

Now, by Equation (D.2), we get

$$\begin{aligned}
& \tau(s'_1,a)\left([\![\chi^C_k]\!]\right) \leq \tau(s'_2,a)\left([\![(\chi^C_k)_\varepsilon]\!]\right) && \\
\implies\ & \tau(s'_1,a)(C_k) \leq \tau(s'_2,a)(C^\varepsilon_k) && \text{for all } k \\
\implies\ & \tau(s'_1,a)(C_{k'}) \leq \lim_{k\to\infty}\tau(s'_2,a)(C^\varepsilon_k) && \text{for a fixed } k' \\
\implies\ & \tau(s'_1,a)(C_{k'}) \leq \tau(s'_2,a)\left(\bigcap_{k\in\mathbb{N}} C^\varepsilon_k\right) && \text{(cont. of measures)} \\
\implies\ & \tau(s'_1,a)([\![\chi^C]\!]) \leq \tau(s'_2,a)(\mathcal{R}(C)) && [\![\chi^C]\!] \subseteq C_{k'} \\
\implies\ & \tau(s'_1,a)(C) \leq \tau(s'_2,a)(\mathcal{R}(C)) && C \subseteq [\![\chi^C]\!].
\end{aligned}$$

Next assume that C is countably infinite and let $\{C_k\}_{k\in\mathbb{N}}$ be an increasing sequence of finite sets such that $\bigcup_{k\in\mathbb{N}} C_k = C$. Since every C_k is finite, we get

$$\tau(s'_1,a)(C_k) \leq \tau(s'_2,a)(\mathcal{R}(C_k))$$

from what we just proved for the finite case. By continuity of measures, this implies

$$\tau(s_1', a)(C) = \tau(s_1', a)\left(\bigcup_{k\in\mathbb{N}} C_k\right) \leq \tau(s_2', a)\left(\bigcup_{k\in\mathbb{N}} \mathcal{R}(C_k)\right) = \tau(s_2', a)(\mathcal{R}(C)).$$

• We now prove the second item.
($\Longrightarrow$) For this we first prove that

$$s_1 \precsim s_2 \quad \text{implies} \quad \forall \varphi \in \mathrm{TML}^{\leq}. s_2 \models \varphi \implies s_1 \models \varphi$$

by induction on φ. We only consider here the cases of $\varphi = m_p t$ and $\varphi = M_p^a \varphi'$, since the other cases are as in the first item.

Case $\varphi = m_p t$: $s_2 \models m_p t$ means that $F_{s_2}(t) \leq p$. Since $s_1 \precsim s_2$, we know that $F_{s_1}(t) \leq F_{s_2}(t)$, so $F_{s_2}(t) \leq p$, implying $s_1 \models m_p t$.

Case $\varphi = M_p^a \varphi'$: $s_2 \models M_p^a \varphi'$ means that $\tau(s_2, a)([\![\varphi']\!]) \leq p$. By induction hypothesis, we know that $[\![\varphi']\!]$ is $\precsim$-closed, and hence we get $\tau(s_1, a)([\![\varphi']\!]) \leq \tau(s_2, a)([\![\varphi']\!])$, so $s_1 \models M_p^a \varphi'$.

We now prove the claim that

$$s_1 \precsim_\varepsilon s_2 \quad \text{implies} \quad \forall \varphi \in \mathrm{TML}^{\leq}. s_2 \models (\varphi)_\varepsilon \implies s_1 \models \varphi$$

by induction on φ. The only case left to consider is the case where $\varphi = M_p^a \varphi'$, since the remaining cases are as before

Case $\varphi = M_p^a \varphi'$: We have $\tau(s_2, a)([\![(\varphi')_\varepsilon]\!]) \leq p$. Now, $\tau(s_2, a)([\![(\varphi')_\varepsilon]\!]) = \tau((s_2)_\varepsilon, a)([\![\varphi']\!])$, and since we now know that $[\![\varphi']\!]$ is $\precsim$-closed, we get

$$p \geq \tau((s_2)_\varepsilon, a)([\![\varphi']\!]) \geq \tau(s_1, a)([\![\varphi']\!]),$$

meaning that $s_1 \models M_r^a \varphi'$.
($\Longleftarrow$) Same as the first item. ■

As a special case of Theorem D.7.1, we have also shown that $\mathrm{TML}^{\geq}$ and $\mathrm{TML}^{\leq}$ characterise simulation for SMDPs. Conceptually, Theorem D.7.1 says that if s_1 ε-simulates s_2, then s_2 satisfies the ε-perturbation of any property that s_2 satisfies for the $\mathrm{TML}^{\geq}$ fragment of TML, and vice versa for the $\mathrm{TML}^{\leq}$ fragment.

By Lemma D.5.1 and Theorem D.7.1, we get the following corollary, connecting our simulation distance with the properties expressible in the logic TML.

Corollary D.7.2. *Let $\varepsilon \in \mathbb{Q}_{\geq 0}$ with $\varepsilon \geq 1$. For finite SMDPs the following holds.*

- $d(s_1, s_2) \leq \varepsilon$ *if and only if* $\forall \varphi \in \mathrm{TML}^{\geq}. s_1 \models \varphi \implies s_2 \models (\varphi)_\varepsilon$.
- $d(s_1, s_2) \leq \varepsilon$ *if and only if* $\forall \varphi \in \mathrm{TML}^{\leq}. s_2 \models (\varphi)_\varepsilon \implies s_1 \models \varphi$.

By Proposition D.2.4, we also get a logical characterisation of bisimulation for SMDPs in terms of TML, which is simpler than the one given in [18, 26].

Theorem D.7.3.

$$s_1 \sim s_2 \quad \textit{if and only if} \quad \forall \varphi \in \mathrm{TML}.\ s_1 \models \varphi \iff s_2 \models \varphi.$$

Proof. ($\Longrightarrow$) We first prove that $s_1 \models \varphi$ implies $s_2 \models \varphi$ for all $\varphi \in \mathrm{TML}$. The proof proceeds by induction on φ. The cases of disjunction and conjunction are standard, and the cases of $\varphi = \alpha$ and $\varphi = \neg\alpha$ are the same as in the proof of Theorem B.6.1.

Case $\varphi = \ell_p t$: $s_1 \models \ell_p t$ means $F_{s_1}(t) \geq p$, and since $F_{s_1}(t) = F_{s_2}(t)$, we get $s_2 \models \ell_p t$.

Case $\varphi = m_p t$: Same argument as $\ell_p t$.

Case $\varphi = L^a_p \varphi'$: $s_1 \models L^a_p \varphi'$ means $\tau(s_1, a)([\![\varphi']\!]) \geq p$. We know that there exists a coupling Δ_a such that

$$\begin{aligned}
\tau(s_1, a)([\![\varphi']\!]) &= \sum_{s \in [\![\varphi']\!]} \tau(s_1, a)(s) \\
&= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in S} \Delta_a(s, s') \\
&= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in [\![\varphi']\!]} \Delta_a(s, s') && \text{(ind. hyp.)} \\
&\leq \sum_{s' \in [\![\varphi']\!]} \tau(s_2, a)(s') \\
&= \tau(s_2, a)([\![\varphi']\!]),
\end{aligned}$$

so $s_2 \models L^a_p \varphi'$.

Case $\varphi = M^a_p \varphi'$: Same argument as $L^a_p \varphi'$.

Next we prove that $s_2 \models \varphi$ implies $s_1 \models \varphi$ for all $\varphi \in \mathrm{TML}$, again by induction on φ. All cases except $\varphi = L^a_p \varphi'$ and $\varphi = M^a_p \varphi'$ are as before.

Case $\varphi = L^a_p \varphi'$: We have $\tau(s_2, a)([\![\varphi']\!]) \geq p$. Since $s_1 \sim s_2$, in particular we have $s_2 \precsim s_1$, so by Theorem D.7.1 and Proposition D.2.3 we get

$$\tau(s_2, a)([\![\varphi']\!]) \leq \tau(s_1, a)([\![\varphi']\!]),$$

and hence $s_1 \models L^a_p \varphi'$.

Case $\varphi = M^a_p \varphi'$: Same argument as $L^a_p \varphi'$.

($\Longleftarrow$) We have assumed that $\forall \varphi \in \mathrm{TML}. s_1 \models \varphi \iff s_2 \models \varphi$, and hence we also get $\forall \varphi \in \mathrm{TML}^{\geq}. s_1 \models \varphi \implies s_2 \models \varphi$ and $\forall \varphi \in \mathrm{TML}^{\leq}. s_2 \models \varphi \implies s_1 \models \varphi$. By Theorem B.6.1 we then get $s_1 \precsim s_2$ and $s_2 \precsim s_1$, and hence Proposition D.2.4 implies $s_1 \sim s_2$. ∎

With the logical characterisation of ε-simulation in hand, we can now prove the promised result that the kernel of the simulation distance is simulation. For this, we first need the following technical lemma. Given a weight function Δ_a, we let

$$\operatorname{supp}(\Delta_a) = \{(s, s') \in S \times S \mid \Delta_a(s, s') > 0\}.$$

Lemma D.7.4. *Assume that $M = (S, \tau, \rho, L)$ is finitely supported and consider $s_1, s_2 \in S$. If $s_1 \precsim_\varepsilon s_2$ for all $\varepsilon > 1$ then for all $\varepsilon > 1$ and $a \in L$ there exists a ε-simulation relation R_ε and a weight function Δ_a such that for any $\varepsilon > \varepsilon' > 1$ there exists a ε'-simulation relation $R_{\varepsilon'}$ with $\operatorname{supp}(\Delta_a) \subseteq R_{\varepsilon'}$.*

Proof. Let $\varepsilon > 1$ and $a \in L$. Because we know that $s_1 \precsim_\varepsilon s_2$, there exists a ε-simulation relation R_ε and a weight function Δ_a with $\operatorname{supp}(\Delta_a) \subseteq R_\varepsilon$. Now let $\varepsilon > \varepsilon_1 > 1$. Because $s_1 \precsim_{\varepsilon_1} s_2$, we know that there exists a ε_1-simulation relation R_{ε_1} and a weight function Δ_a^1 such that $\operatorname{supp}(\Delta_a^1) \subseteq R_{\varepsilon_1}$.

If $\operatorname{supp}(\Delta_a) = \operatorname{supp}(\Delta_a^1)$, we are done. If not, it may be the case that for any $\varepsilon_1 > \varepsilon_2 > 1$ there exists a ε_2-simulation relation R_{ε_2} and a weight function Δ_a^2 such that $\operatorname{supp}(\Delta_a^2) \subseteq R_{\varepsilon_2}$. This would imply by monotonicity that for all $\varepsilon_1 > 1$ there exists a ε_1-simulation relation and a weight function Δ_a^1 such that for any $\varepsilon_1 > \varepsilon_2 > 1$ there exists a ε_2-simulation relation R_{ε_2} such that $\operatorname{supp}(\Delta_a^1) \subseteq R_{\varepsilon_2}$, in which case we are also done.

If this is not the case, then there must exist some $\varepsilon_1 > \varepsilon_2 > 1$ such that $\operatorname{supp}(\Delta_a^1) \not\subseteq R_{\varepsilon_2}$ for any ε_2-simulation relations R_{ε_2}. However, we know that $s_1 \precsim_{\varepsilon_2} s_2$, so there exists a ε_2-simulation relation R_{ε_2} and a weight function Δ_a^2 such that $\operatorname{supp}(\Delta_a^2) \subseteq R_{\varepsilon_2}$. Note that we must have $\operatorname{supp}(\Delta_a^1) \neq \operatorname{supp}(\Delta_a^2)$ because we have $\operatorname{supp}(\Delta_a^1) \not\subseteq R_{\varepsilon_2}$ but $\operatorname{supp}(\Delta_a^2) \subseteq R_{\varepsilon_2}$. If $\operatorname{supp}(\Delta_a) = \operatorname{supp}(\Delta_a^2)$, we are done. If not, it may be the case that for any $\varepsilon_2 > \varepsilon_3 > 1$ there exists a ε_3-simulation relation R_{ε_3} with $\operatorname{supp}(\Delta_a^2) \subseteq R_{\varepsilon_3}$. This would by monotonicity again imply that we are done.

If this is not the case, then there must exist some $\varepsilon_2 > \varepsilon_3 > 1$ such that $\operatorname{supp}(\Delta_a^2) \not\subseteq R_{\varepsilon_3}$ for all ε_3-simulation relations R_{ε_3}. However, we know that $s_1 \precsim_{\varepsilon_3} s_2$, so there exists a ε_3-simulation relation R_{ε_3} and a weight function Δ_a^3 such that $\operatorname{supp}(\Delta_a^3) \subseteq R_{\varepsilon_3}$. Note that $\operatorname{supp}(\Delta_a^2) \neq \operatorname{supp}(\Delta_a^3)$ and also $\operatorname{supp}(\Delta_a^1) \neq \operatorname{supp}(\Delta_a^3)$ since R_{ε_3} is also a ε_2-simulation relation. If $\operatorname{supp}(\Delta_a) = \operatorname{supp}(\Delta_a^3)$, we are done. If not, it may be the case that for any $\varepsilon_3 > \varepsilon_4 > 1$ there exists a ε_4-simulation relation R_{ε_4} with $\operatorname{supp}(\Delta_a^3) \subseteq R_{\varepsilon_4}$ which, by monotonicity, would imply that we are done.

Continuing in this way, we get a sequence

$$\operatorname{supp}(\Delta_a^1), \operatorname{supp}(\Delta_a^2), \operatorname{supp}(\Delta_a^3), \ldots,$$

all pairwise different from each other. However, since

$$\Delta_a(s, s') > 0 \quad \text{implies} \quad \tau(s_1, a)(s) > 0 \text{ and } \tau(s_2, a)(s') > 0,$$

$\mathrm{supp}(\Delta_a)$ seen as a function of Δ_a can only take on finitely many values. Hence the process must eventually stop and we find witnesses for the statement of the lemma. ■

Theorem D.7.5.

$$s_1 \precsim s_2 \quad \textit{implies} \quad d(s_1, s_2) = 1.$$

For finitely supported SMDPs, it also holds that

$$d(s_1, s_2) = 1 \quad \textit{implies} \quad s_1 \precsim s_2.$$

Proof. The first point is immediate: If $s_1 \precsim s_2$, this means that $s_1 \precsim_1 s_2$, so $d(s_1, s_2) = 1$.

For the second point, assume that $d(s_1, s_2) = 1$. This means that either $s_1 \precsim_1 s_2$, in which case we are done, or $s_1 \precsim_\varepsilon s_2$ for all $\varepsilon > 1$, in which case we wish to prove that this implies that

$$s_1 \models \varphi \implies s_2 \models \varphi \quad \text{for all } \varphi \in \mathrm{TML}^{\geq}. \tag{D.3}$$

This would imply, by Theorem D.7.1, that $s_1 \precsim s_2$, and we are done. Hence we now prove the claim in (D.3) by induction on φ.

($\varphi = \alpha$ or $\varphi = \neg\alpha$): Choose some $\varepsilon > 1$. Then there exists a ε-simulation relation R such that $s_1 R s_2$. This implies that $L(s_1) = L(s_2)$, so if $s_1 \models \varphi$, then also $s_2 \models \varphi$.

($\varphi = \varphi_1 \vee \varphi_2$): If $s_1 \models \varphi_1 \vee \varphi_2$, then $s_1 \models \varphi_1$ or $s_1 \models \varphi_2$. By induction hypothesis, this implies that $s_2 \models \varphi_1$ or $s_2 \models \varphi_2$, so $s_2 \models \varphi_1 \vee \varphi_2$.

($\varphi = \varphi_1 \wedge \varphi_2$): Similar to the case $\varphi = \varphi_1$.

($\varphi = \ell_p t$): Assume $s_1 \models \ell_p t$, which means that $F_{s_1}(t) \geq p$. Assume towards a contradiction that $F_{s_2}(t) < p$, and let $\varepsilon' = p - F_{s_2}(t) > 0$. Then there exists $\delta > 0$ such that for any $t < x < t + \delta$ we have $F_{s_2}(x) - F_{s_2}(t) < \varepsilon'$. If $t = 0$ then for any $\varepsilon > 0$ we have

$$p > F_{s_2}(t) = F_{s_2}(0) = F_{s_2}(\varepsilon \cdot t) \geq F_{s_1}(t) \geq p,$$

which is a contradiction. If $t > 0$, then choose an $\varepsilon > 0$ such that $1 < \varepsilon < \frac{t+\delta}{t}$, meaning that $t < \varepsilon \cdot t < t + \delta$. By right-continuity, this implies that $F_{s_2}(\varepsilon \cdot t) - F_{s_2}(t) < \varepsilon'$. Hence we get

$$p > F_{s_2}(t) \geq F_{s_2}(\varepsilon \cdot t) \geq F_{s_1}(t) \geq p,$$

which is also a contradiction.

($\varphi = L^a_p \varphi'$): Assume $s_1 \models L^a_p \varphi'$, meaning that $\tau(s_1, a)([\![\varphi']\!]) \geq p$, and choose some $\varepsilon > 1$. By Lemma D.7.4, there exists a ε-simulation relation R_ε and a coupling Δ_a with $\mathrm{supp}(\Delta_a) \subseteq R_\varepsilon$ such that for any $\varepsilon > \varepsilon' > 1$ there

exists a ε'-simulation relation $R_{\varepsilon'}$ with $\mathrm{supp}(\Delta_a) \subseteq R'_\varepsilon$. We then get

$$\begin{aligned} p &\leq \tau(s_1, a)([\![\varphi']\!]) \\ &= \sum_{s \in [\![\varphi']\!]} \tau(s_1, a)(s) \\ &= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in S} \Delta_a(s, s') \\ &= \sum_{(s,s') \in ([\![\varphi']\!] \times S) \cap \mathrm{supp}(\Delta_a)} \Delta_a(s, s'). \end{aligned}$$

Now, we know that for any $\varepsilon > \varepsilon' > 1$ there exists a ε'-simulation relation $R_{\varepsilon'}$ such that $\mathrm{supp}(\Delta_a) \subseteq R_{\varepsilon'}$. This means that for any $(s, s') \in ([\![\varphi']\!] \times S) \cap \mathrm{supp}(\Delta_a)$ we have $s \precsim_{\varepsilon'} s'$. By monotonicity, we therefore get that $s \precsim_{\varepsilon'} s'$ for any $\varepsilon' > 1$. The induction hypothesis then gives

$$\begin{aligned} &\sum_{(s,s') \in ([\![\varphi']\!] \times S) \cap \mathrm{supp}(\Delta_a)} \Delta_a(s, s') \\ &= \sum_{(s,s') \in ([\![\varphi']\!] \times [\![\varphi']\!]) \cap \mathrm{supp}(\Delta_a)} \Delta_a(s, s') \\ &= \sum_{s \in [\![\varphi']\!]} \sum_{s' \in [\![\varphi']\!]} \Delta_a(s, s') \\ &\leq \sum_{s \in S} \sum_{s' \in [\![\varphi']\!]} \Delta_a(s, s') \\ &= \sum_{s' \in [\![\varphi']\!]} \tau(s_2, a)(s') \\ &= \tau(s_2, a)([\![\varphi']\!]), \end{aligned}$$

which implies that $s_2 \models L^a_p \varphi'$. ■

D.7.1 Reachability Properties

We will now argue that the simulation distance behaves nicely also with respect to linear-time properties, by proving preservation of reachability properties up to perturbations.

The probability of reaching a given set of states in an SMDP depends on the choice of actions in each state. The non-determinism introduced by this choice is typically resolved by means of *schedulers*. Here we consider probabilistic schedulers σ of type $S^* \to \mathcal{D}(A)$, telling us what the probability is of selecting an action $a \in A$ depending on the history of the states visited so far.

Given a SMDP $M = (S, \tau, \rho, L)$, a *path* in M is a sequence

$$\pi = (s_1, t_1), (s_2, t_2), \ldots,$$

where $s_i \in S$ and $t_i \in \mathbb{R}_{\geq 0}$. Intuitively, a path π denotes an execution of the SMDP, where s_i denotes the ith state visited, and t_i denotes the time spent in s_i. We denote by $\Pi(M)$ the set of all paths in M, and we let $\pi[i] = s_i$ and $\pi\langle i \rangle = t_i$.

Let $X \subseteq S$. Then

$$\Diamond^t X = \{\pi \in \Pi(M) \mid \exists i \in \mathbb{N}.\pi[i] \in X \text{ and } \sum_{j=1}^{i-1} \pi\langle j \rangle \leq t\}$$

is the set of paths that eventually reach a state in X and does so within time t.

Given a scheduler σ, we define a probability

$$\mathbb{P}_s^\sigma(S_1) = \sum_{a \in A} \sum_{s' \in S} \tau^\sigma(s,a)(s') \cdot \rho(s)$$
$$\mathbb{P}_s^\sigma(S_1, S_2, \ldots, S_n) = \sum_{a \in A} \sum_{s' \in S} \tau^\sigma(s,a)(s') \cdot (\rho(s) * \mathbb{P}_{s'}^\sigma(S_2, \ldots, S_n))$$

through the usual cylinder construction. Then $\mathbb{P}_s^\sigma(S_1, \ldots, S_n)(t)$ is the probability, starting from s and under the scheduler σ, to first visit a state in S_1, then a state in S_2, and so on, until a state in S_n is reached, and the total time elapsed is at most t.

Lemma D.7.6. *Let β be a Boolean combination of atomic propositions. If $s_1 \precsim_\varepsilon s_2$, then for any scheduler σ there exists a scheduler σ' such that*

$$\mathbb{P}_{s_1}^\sigma(\underbrace{[\![\beta]\!]^c, \ldots, [\![\beta]\!]^c}_{n-1 \text{ times}}, [\![\beta]\!])(t) \leq \mathbb{P}_{s_2}^{\sigma'}(\underbrace{[\![\beta]\!]^c, \ldots, [\![\beta]\!]^c}_{n-1 \text{ times}}, [\![\beta]\!])(\varepsilon \cdot t)$$

for all $n \in \mathbb{N}$ and $t \in \mathbb{R}_{\geq 0}$.

Proof. Let R be a ε-simulation relation witnessing that $s_1 \precsim_\varepsilon s_2$.

Case $n = 1$: For each $a \in A$ there exists a coupling Δ_a such that

$$\begin{aligned}\mathbb{P}_{s_1}^\sigma([\![\beta]\!])(t) &= \sum_{a \in A} \sum_{s \in [\![\beta]\!]} \tau(s_1,a)(s) \cdot \sigma(s_1)(a) \cdot \rho(s_1)(t) \\ &= \sum_{a \in A} \sum_{s \in [\![\beta]\!]} \sum_{s' \in S} \Delta_a(s,s') \cdot \sigma(s_1)(a) \cdot \rho(s_1)(t).\end{aligned}$$

If $s \in [\![\beta]\!]$ and $s' \notin [\![\beta]\!]$, then $s \not\precsim_\varepsilon s'$, and hence $(s,s') \notin R$, so $\Delta_a(s,s') = 0$. We

therefore get

$$\begin{aligned}\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!])(t) &= \sum_{a\in A}\sum_{s\in[\![\beta]\!]}\sum_{s'\in[\![\beta]\!]}\Delta_a(s,s')\cdot\sigma(s_1)(a)\cdot\rho(s_1)(t)\\ &= \sum_{a\in A}\sum_{s'\in[\![\beta]\!]}\sum_{s\in[\![\beta]\!]}\Delta_a(s,s')\cdot\sigma(s_1)(a)\cdot\rho(s_1)(t)\\ &\leq \sum_{a\in A}\sum_{s'\in[\![\beta]\!]}\sum_{s\in S}\Delta_a(s,s')\cdot\sigma(s_1)(a)\cdot\rho(s_1)(t)\\ &= \sum_{a\in A}\sum_{s'\in[\![\beta]\!]}\tau(s_2,a)(s')\cdot\sigma(s_1)(a)\cdot\rho(s_1)(t).\end{aligned}$$

Now we define $\sigma'(s_2)(a) = \sigma(s_1)(a)$ and observe that $\rho(s_1)(t) \leq \rho(s_2)(\varepsilon\cdot t)$ since we have assumed $s_1 \precsim_\varepsilon s_2$. Hence we get

$$\begin{aligned}\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!])(t) &\leq \sum_{a\in A}\sum_{s'\in[\![\beta]\!]}\tau(s_2,a)(s')\cdot\sigma'(s_2)(a)\cdot\rho(s_2)(\varepsilon\cdot t)\\ &= \mathbb{P}^{\sigma'}_{s_2}([\![\beta]\!])(\varepsilon\cdot t).\end{aligned}$$

Case $n > 1$: For any $a \in A$ we again get a coupling Δ_a such that

$$\begin{aligned}&\mathbb{P}^{\sigma}_{s_1}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-1\text{ times}},[\![\beta]\!])(t)\\ &= \sum_{a\in A}\sum_{s\in[\![\beta]\!]^c}\tau(s_1,a)(s)\cdot\sigma(s_1)(a)\cdot(\rho(s_1)*\mathbb{P}^{\sigma}_{s}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!]))(t)\\ &= \sum_{a\in A}\sum_{s\in[\![\beta]\!]^c}\sum_{s'\in S}\Delta_a(s,s')\cdot\sigma(s_1)(a)\cdot(\rho(s_1)*\mathbb{P}^{\sigma}_{s}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!]))(t)\\ &= \sum_{a\in A}\sum_{s\in[\![\beta]\!]^c}\sum_{s'\in[\![\beta]\!]^c}\Delta_a(s,s')\cdot\sigma(s_1)(a)\cdot(\rho(s_1)*\mathbb{P}^{\sigma}_{s}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!]))(t).\end{aligned}$$

Since $s \not\precsim_\varepsilon s'$ implies $\Delta_a(s,s') = 0$, any term where $s \not\precsim_\varepsilon s'$ contributes nothing to the sum. Hence we may assume that $s \precsim_\varepsilon s'$. By induction hypothesis, we then get that for any s' there exists $\sigma'_{s'}$ such that

$$\mathbb{P}^{\sigma}_{s}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!])(t) \leq \mathbb{P}^{\sigma'_{s'}}_{s'}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!])(\varepsilon\cdot t).$$

Now let $w \in S^*$ and define

$$\sigma''(s'w)(a) = \sigma'_{s'}(w)(a) \quad \text{and} \sigma''(s_2)(a) = \sigma(s_1)(a).$$

By Proposition D.3.8 we get

$$(\rho(s_1)*\mathbb{P}^{\sigma}_{s}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!]))(t) \leq (\rho(s_2)*\mathbb{P}^{\sigma'_{s'}}_{s'}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2\text{ times}},[\![\beta]\!]))(\varepsilon\cdot t).$$

Hence we get

$$\begin{aligned}
&\mathbb{P}^{\sigma}_{s_1}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-1 \text{ times}},[\![\beta]\!])(t)\\
&\leq \sum_{a\in A}\sum_{s'\in[\![\beta]\!]^c}\tau(s_2,a)(s')\cdot\sigma''(s_2)(a)\cdot(\rho(s_2)*\mathbb{P}^{\sigma''}_{s'}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-2 \text{ times}},[\![\beta]\!]))(\varepsilon\cdot t)\\
&= \mathbb{P}^{\sigma''}_{s_2}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-1 \text{ times}},[\![\beta]\!])(\varepsilon\cdot t). \qquad \blacksquare
\end{aligned}$$

Given our notion of ε-simulation, we can prove the following result.

Theorem D.7.7. *Let β be a Boolean combination of atomic propositions. If we have $s_1 \precsim_\varepsilon s_2$, then for any scheduler σ there exists a scheduler σ' such that*

$$\mathbb{P}^{\sigma}_{s_1}(\Diamond^t[\![\beta]\!]) \leq \mathbb{P}^{\sigma'}_{s_2}(\Diamond^{\varepsilon\cdot t}[\![\beta]\!]) \quad (\textit{or equivalently, } \mathbb{P}^{\sigma}_{s_1}(\neg\Diamond^t[\![\beta]\!]) \geq \mathbb{P}^{\sigma'}_{s_2}(\neg\Diamond^{\varepsilon\cdot t}[\![\beta]\!])).$$

Proof. First note that for any s and σ, we have

$$\mathbb{P}^{\sigma}_{s}(\Diamond^t[\![\beta]\!]) = \sum_{n\in\mathbb{N}}\mathbb{P}^{\sigma}_{s}([\![\beta]\!]^t_n),$$

where

$$[\![\beta]\!]^t_n = \{\pi\in\Pi(M) \mid \pi[n]\in[\![\beta]\!], \forall k<n.\pi[k]\notin[\![\beta]\!], \text{ and } \sum_{j=1}^{n-1}\pi\langle j\rangle \leq t\}.$$

We will now argue that for any σ there exists σ' such that

$$\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!]^t_n) \leq \mathbb{P}^{\sigma'}_{s_2}([\![\beta]\!]^{\varepsilon\cdot t}_n)$$

for any $n\in\mathbb{N}$ and $t\in\mathbb{R}_{\geq 0}$.

Case $n=1$**:** In this case we have $\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!]^t_n) = \mathbb{1}_{[\![\beta]\!]}(s_1)$ and $\mathbb{P}^{\sigma'}_{s_2}([\![\beta]\!]^{\varepsilon\cdot t}_n) = \mathbb{1}_{[\![\beta]\!]}(s_2)$. Since $s_1 \precsim_\varepsilon s_2$, we get $s_1\in[\![\beta]\!]$ if and only if $s_2\in[\![\beta]\!]$, and hence $\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!]^t_n) = \mathbb{P}^{\sigma'}_{s_2}([\![\beta]\!]^{\varepsilon\cdot t}_n)$ for any σ and σ'.

Case $n>1$**:** In this case we have

$$\mathbb{P}^{\sigma}_{s_1}([\![\beta]\!]^t_n) = \mathbb{1}_{[\![\beta]\!]^c}(s_1)\cdot\mathbb{P}^{\sigma}_{s_1}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-1 \text{ times}},[\![\beta]\!])(t)$$

and

$$\mathbb{P}^{\sigma'}_{s_2}([\![\beta]\!]^{\varepsilon\cdot t}_n) = \mathbb{1}_{[\![\beta]\!]^c}(s_2)\cdot\mathbb{P}^{\sigma'}_{s_2}(\underbrace{[\![\beta]\!]^c,\ldots,[\![\beta]\!]^c}_{n-1 \text{ times}},[\![\beta]\!])(\varepsilon\cdot t).$$

Since $s_1 \precsim_\varepsilon s_2$, we have $\mathbb{1}_{[\![\beta]\!]^c}(s_1) = \mathbb{1}_{[\![\beta]\!]^c}(s_2)$. The result then follows from Lemma D.7.6. $\blacksquare$

Note that the above result might find useful applications for speeding up the computation time required by model checking tools to disprove certain types of reachability properties. For example, consider the atomic proposition bad, identifying all the states considered "not safe" in the SMDP. Usually, given a process s, one wants to verify that, under all possible schedulers σ, the probability $\mathbb{P}_s^\sigma(\neg\Diamond^t[\![\mathsf{bad}]\!])$ is above a certain threshold value $\delta \leq 1$, meaning that the SMDP is unlikely to end up in an unsafe configuration within a time horizon bounded by t. Then, to disprove this property one only needs to provide a scheduler σ' and a process s' such that $s' \precsim_\varepsilon s$ and $\mathbb{P}_{s'}^{\sigma'}(\neg\Diamond^{\frac{t}{\varepsilon}}[\![\mathsf{bad}]\!]) < \delta$. Indeed, given that $s' \precsim_\varepsilon s$, by Theorem D.7.7

$$\mathbb{P}_{s'}^{\sigma'}\left(\neg\Diamond^{\frac{t}{\varepsilon}}[\![\mathsf{bad}]\!]\right) < \delta \overset{Th.D.7.7}{\Longrightarrow} \exists\sigma.\ \mathbb{P}_s^\sigma(\neg\Diamond^t[\![\mathsf{bad}]\!]) < \delta.$$

Since s simulates s', s' can be thought of as a simplified abstraction of s, which is usually a smaller process. Hence, finding a scheduler σ' for s' which gives a counterexample may be much simpler than finding one for s. Moreover, the above technique is robust to perturbations of ε.

D.8 The Topology of TML

A common practice in science and engineering is that of approximating and refining models. We would therefore like to ensure that whenever we make better and better approximations of a model, whatever property holds for the approximations should also hold for the model that is approximated. In our terms, this means that we would like the sets satisfying formulas in TML to be *closed*, because then we would know that whenever a sequence of states $\{s_n\}$ which converges to some state s satisfies some property of TML, then s also satisfies that property.

Since the concept of open and closed sets are topological concepts, we introduce the topology generated by our distance. Note that the concept of closed set and sequentially closed set need not coincide in arbitrary topological spaces. However, they do coincide for hemimetric spaces, since these are first-countable [12]. Moreover, hemimetric spaces are not in general Hausdorff, so limits need not be unique.

Because the distance is non-symmetric, we can in fact generate two different topologies. For $r > 1$, the open balls of the form

$$\mathcal{B}_r^L(s) = \{s' \mid d(s, s') < r\}$$

generate the *left-centered topology* and open balls of the form

$$\mathcal{B}_r^R(s) = \{s' \mid d(s', s) < r\}$$

generate the *right-centered topology*. These two topologies behave differently, as we will now show.

Lemma D.8.1. *The following holds in the right-centered topology.*

1. $[\![\ell_p t]\!]$ *is closed.*
2. *If* $p = 0$, *then* $[\![\ell_p t]\!]$ *is open.*
3. *If* $p > 0$, *then* $[\![\ell_p t]\!]$ *is not open.*
4. *If* $p = 1$, *then* $[\![m_p t]\!]$ *is closed.*
5. *If* $p < 1$, *then* $[\![m_p t]\!]$ *is not closed.*

Proof. 1. Let $\{s_k\}$ be a sequence of states such that $s_k \in [\![\ell_p t]\!]$ for all k and $\lim_{k\to\infty} s_k \ni s$. We must show that $s \in [\![\ell_p t]\!]$. Assume towards a contradiction that $s \notin [\![\ell_p t]\!]$ and let $\varepsilon > 1$. First note that if $t = 0$, then $F_{s_k}(0) \geq p$ for all k. Since $\lim_{k\to\infty} s_k \ni s$, there must exist some n such that $d(s_n, s) < \varepsilon$. But then

$$p \leq F_{s_n}(0) \leq F_s(\varepsilon \cdot 0) = F_s(0) < p,$$

which is a contradiction.

We can therefore now assume that $t > 0$. Let $p - F_s(t) = \varepsilon > 0$. By right-continuity, there exists a $\delta > 0$ such that $x < c < x + \delta$ implies $|F_s(x) - F_s(c)| < \varepsilon$. Now choose ε' such that $1 < \varepsilon' < \frac{t+\delta}{t}$, which means that $t < \varepsilon' \cdot t < t + \delta$. Then we get

$$|F_s(t) - F_s(\varepsilon' \cdot t)| < \varepsilon$$

which implies $F_s(t) \leq F_s(\varepsilon' \cdot t) < p$. However, since $\lim_{k\to\infty} s_k \ni s$, we know that there must exist some n such that

$$p \leq F_{s_n}(t) \leq F_s(\varepsilon' \cdot t) < p,$$

which is a contradiction.

2. If $p = 0$, then $F_s(t) \geq p$ for any s, so $s \in [\![\ell_p t]\!]$ for any s.

3. If $p > 0$, let $F_s = \mathit{Unif}[a, t]$ and $F_{s'} = \mathit{Unif}[t, b]$ for some a and b (if $t = 0$, $F_s = \delta_0$ instead). Then $F_s(t) = 1 \geq p$, so $s \in [\![\ell_p t]\!]$, but $F_{s'}(t) = 0 < p$, and hence $s' \notin [\![\ell_p t]\!]$. However, for any $r > 1$, we must have $s' \in \mathcal{B}_r^R(u)$, since $F_s(t) \geq F_{s'}(t)$ for any t. Hence $\mathcal{B}_r^R(s) \not\subseteq [\![\ell_p t]\!]$ for any r.

4. If $p = 1$, then $F_s(t) \leq p$ for any s, so $s \in [\![m_p t]\!]$ for any s.

5. If $p < 1$, let s be a state with $F_s = \mathit{Unif}[a, b]$ and let $\{s_k\}$ be a sequence of states such that $F_{s_k} = \mathit{Unif}[b, c]$ for any k. Then $\lim_{k\to\infty} s_k \ni s$ and $F_{s_k}(b) = 0 \leq p$, so $s_k \in [\![m_p t]\!]$. However, $p < 1 = F_s(b)$, so $u \notin [\![m_p t]\!]$. ∎

Lemma D.8.2. *The following holds in the left-centered topology.*

1. *If $p = 1$, then $[\![m_p t]\!]$ is open.*
2. *If $p < 1$, then $[\![m_p t]\!]$ is not open.*
3. *If $p = 0$, $[\![\ell_p t]\!]$ is closed.*
4. *If $p > 0$, $[\![\ell_p t]\!]$ is not closed.*

Proof. 1. If $p = 1$, then $F_s(t) \leq p$ for any s.

2. If $p < 1$, then let $F_{s'} = \mathit{Unif}[a, t]$ and $F_s = \mathit{Unif}[t, b]$ for some a and b (if $t = 0$, let $F_{s'} = \delta_0$ instead). Then $F_s(t) = 0 \leq p$ and hence $s \in [\![m_p t]\!]$, but $F_{s'}(t) = 1 > p$, so $s' \notin [\![m_p t]\!]$. However, $s' \in \mathcal{B}^L_r(u)$ for any $r > 1$ because $F_{s'}(t) \geq F_s(t)$ for all t. Hence $\mathcal{B}^L_r(s) \not\subseteq [\![m_p t]\!]$ for any r.

3. If $p = 0$, then $F_s(t) \geq p$ for any s.

4. If $p > 0$, let s be a state such that $F_s = \mathit{Unif}[b, c]$ and let $\{s_k\}$ be a sequence of states such that $F_{s_k} \sim \mathit{Unif}[a, b]$ for any k. Then $\lim_{k\to\infty} s_k \ni s$ and $F_{s_k}(b) = 1 \geq p$, and hence $s_k \in [\![\ell_p t]\!]$ for any k. However, $p > 0 = F_s(b)$, so $s \notin [\![\ell_p t]\!]$. ■

The case of $[\![m_p t]\!]$ is missing in Lemma D.8.2. We have not been able to determine whether $[\![m_p t]\!]$ is closed in the left-centered topology, but we strongly suspect that this is the case. Hence we have the following conjecture.

Conjecture D.8.3. *In the left-centered topology, $[\![m_p t]\!]$ is closed.*

We can now show that approximate reasoning in $\mathrm{TML}^{\geq}$ is sound with respect to the right-centered topology, in the sense that if we have a sequence of states $s_1, s_2, \ldots$ that approximate some state s better and better, then if a property holds for all $s_1, s_2, \ldots$, it will also hold for s.

Theorem D.8.4. *For any $\varphi \in \mathrm{TML}^{\geq}$, $[\![\varphi]\!]$ is closed in the right-centered topology.*

Proof. Let $\varphi \in \mathrm{TML}^{\geq}$. We prove by induction on φ that $[\![\varphi]\!]$ is closed in the right-centered topology.

Case $\varphi = \ell_p t$: This follows by Lemma D.8.1.

Case $\varphi = L^a_p \varphi'$: Let $\{s_k\}$ be a sequence of states such that $\lim_{k\to\infty} s_k \ni s$ and $s_k \models L^a_p \varphi'$. Let $\varepsilon > 1$. Then there exists k' such that $s_{k'} \precsim_\varepsilon s$, and by assumption, $\tau(s_{k'}, a)([\![\varphi']\!]) \geq p$. By Theorem D.7.1, this implies $\tau(s, a)([\![(\varphi')_\varepsilon]\!]) \geq p$. Since this holds for any $\varepsilon > 1$ and $\lim_{\varepsilon\to 1}(\varphi')_\varepsilon = \varphi'$, we then get

$$\tau(s, a)([\![\varphi']\!]) \geq p.$$

Case $\varphi = \varphi_1 \wedge \varphi_2$: Since a finite intersection of closed sets is again closed, we get that $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$ is closed by the induction hypothesis.

Case $\varphi = \varphi_1 \vee \varphi_2$: Since a finite union of closed sets is again closed, we get that $[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$ is closed by the induction hypothesis. ■

Furthermore, if we assume Conjecture D.8.3, then we can use a symmetric argument as in the proof of Theorem D.8.4 to show that approximate reasoning in $\mathrm{TML}^{\leq}$ is sound with respect to the left-centered topology.

Conjecture D.8.5. *For any $\varphi \in \mathrm{TML}^{\leq}$, $[\![\varphi]\!]$ is closed in the left-centered topology.*

D.9 Conclusion and Open Problems

We have proposed a quantitative extension of the notion of simulation relation on SMDPs, called ε-simulation, comparing the relative speed of different processes. This quantitative notion of simulation relation induces a multiplicative hemimetric, which we call simulation distance, measuring the least acceleration factor needed by a process to speed up its actions in order to behave at least as fast as another process.

We have given an efficient algorithm to compute the simulation distance and identified a class of distributions for which the algorithm works on finite SMDPs. Furthermore, we have shown that, under mild conditions on the composition of residence-time distributions on states, a generalised version of CSP-like parallel composition on SMDPs is non-expansive with respect to this distance, showing that our distance is suitable for compositional reasoning. We have also shown the connection between our distance and properties expressible in a timed extension of Markovian logic. Namely, we have shown that if the simulation distance between s_1 and s_2 is at most ε, then s_1 satisfies the ε-perturbation of any property that s_2 satisfies. This result also gives a novel logical characterisation of simulation and bisimulation for semi-Markov decision processes. Lastly we have investigated the topological properties of our distance for this logic, and shown that approximate reasoning is sound in the limit.

Instead of using the usual stochastic order to relate the timing behaviour of states as we have done, one could also consider many other kinds of stochastic orders, for example ones that compare the expected value of the distributions. This may be more natural for applications where one wants to consider an exponential distribution with a high enough rate to be faster than a uniform distribution.

We have shown that the timing distributions that are obtained when composing systems are compatible with the algorithm for computing the distance only in the case when composing systems either on the left or on the right. A more general result showing that this also happens when composing on both sides an arbitrary number of components seems difficult. Nonetheless,

we are confident that such a result can be obtained for any concrete case involving common types of distributions used in the literature.

Acknowledgments.

We thank the anonymous reviewers for helpful suggestions as well as Robert Furber and Giovanni Bacci for insightful discussions. This research was supported by the Danish FTP project ASAP, the ERC Advanced Grant LASSO, and the Sino-Danish Basic Research Center IDEA4CPS.

D.10 References

[1] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Additive and multiplicative notions of leakage, and their capacities," in *IEEE 27th Computer Security Foundations Symposium, CSF 2014, Vienna, Austria, 19-22 July, 2014*. IEEE Computer Society, 2014, pp. 308–322. [Online]. Available: https://doi.org/10.1109/CSF.2014.29

[2] C. Baier, B. Engelen, and M. E. Majster-Cederbaum, "Deciding bisimilarity and similarity for probabilistic processes," *J. Comput. Syst. Sci.*, vol. 60, no. 1, pp. 187–231, 2000. [Online]. Available: http://dx.doi.org/10.1006/jcss.1999.1683

[3] C. Baier, J. Katoen, H. Hermanns, and V. Wolf, "Comparative branching-time semantics for Markov chains," *Inf. Comput.*, vol. 200, no. 2, pp. 149–214, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2005.03.001

[4] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*, ser. Lecture Notes in Computer Science, E. D. Cristofaro and M. K. Wright, Eds., vol. 7981. Springer, 2013, pp. 82–102. [Online]. Available: https://doi.org/10.1007/978-3-642-39077-7_5

[5] D. Chen, F. van Breugel, and J. Worrell, "On the complexity of computing probabilistic bisimilarity," in *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, ser. Lecture Notes in Computer Science, L. Birkedal, Ed., vol. 7213. Springer, 2012, pp. 437–451. [Online]. Available: https://doi.org/10.1007/978-3-642-28729-9_29

[6] E. M. Clarke, D. E. Long, and K. L. McMillan, "Compositional model checking," in *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 1989, pp. 353–362. [Online]. Available: https://doi.org/10.1109/LICS.1989.39190

[7] J. Desharnais, "Logical characterization of simulation for labelled Markov chains," in *Proceedings of the 2nd International Workshop on Probabilistic Methods in Verification*. University of Birmingham, Technical Report, CS-99-8, August 1999, pp. 33–48.

[8] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled Markov processes," *Theor. Comput. Sci.*, vol. 318, no. 3, pp. 323–354, 2004. [Online]. Available: https://doi.org/10.1016/j.tcs.2003.09.013

[9] N. Ferns, P. Panangaden, and D. Precup, "Bisimulation metrics for continuous Markov decision processes," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1662–1714, 2011. [Online]. Available: https://doi.org/10.1137/10080484X

[10] A. Giacalone, C.-C. Jou, and S. A. Smolka, "Algebraic reasoning for probabilistic concurrent systems," in *Proc. IFIP TC2 Working Conference on Programming Concepts and Methods*. North-Holland, 1990, pp. 443–458.

[11] N. Götz, U. Herzog, and M. Rettelbach, "Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras," in *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93, Santa Clara, CA, USA, May 10-14, 1993*, ser. Lecture Notes in Computer Science, L. Donatiello and R. D. Nelson, Eds., vol. 729. Springer, 1993, pp. 121–146.

[12] J. Goubault-Larrecq, *Non-Hausdorff Topology and Domain Theory - Selected Topics in Point-Set Topology*, ser. New Mathematical Monographs. Cambridge University Press, 2013, vol. 22.

[13] H. Hermanns, *Interactive Markov Chains: The Quest for Quantified Quality*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2428. [Online]. Available: https://doi.org/10.1007/3-540-45804-2

[14] H. Hermanns, U. Herzog, and V. Mertsiotakis, "Stochastic process algebras - between LOTOS and Markov chains," *Computer Networks*, vol. 30, no. 9-10, pp. 901–924, 1998.

[15] J. Hillston, *A compositional approach to performance modelling*, ser. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996. [Online]. Available: https://doi.org/10.1017/CBO9780511569951

[16] C. Jou and S. A. Smolka, "Equivalences, congruences, and complete axiomatizations for probabilistic processes," in *CONCUR '90, Theories of Concurrency: Unification and Extension, Amsterdam, The Netherlands, August 27-30, 1990, Proceedings*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. W. Klop, Eds., vol. 458. Springer, 1990, pp. 367–383. [Online]. Available: https://doi.org/10.1007/BFb0039071

[17] D. Kozen, R. Mardare, and P. Panangaden, "Strong completeness for Markovian logics," in *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, K. Chatterjee and J. Sgall, Eds., vol. 8087. Springer, 2013, pp. 655–666. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40313-2_58

[18] M. R. Neuhäußer and J. Katoen, "Bisimulation and logical preservation for continuous-time Markov decision processes," in *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Caires and V. T. Vasconcelos, Eds., vol. 4703. Springer, 2007, pp. 412–427. [Online]. Available: https://doi.org/10.1007/978-3-540-74407-8

[19] P. Panangaden, *Labelled Markov Processes*. Imperial College Press, 2009.

[20] M. Perman, A. Senegacnik, and M. Tuma, "Semi-Markov models with an application to power-plant reliability analysis," *IEEE Transactions on Reliability*, vol. 46, no. 4, pp. 526–532, Dec 1997.

[21] A. Pievatolo, E. Tironi, and I. Valade, "Semi-Markov processes for power system reliability assessment with application to uninterruptible power supply," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1326–1333, Aug 2004.

[22] J. Sack and L. Zhang, "A general framework for probabilistic characterizing formulae," in *Verification, Model Checking, and Abstract Interpretation - 13th International Conference, VMCAI 2012, Philadelphia, PA, USA, January 22-24, 2012. Proceedings*, ser. Lecture Notes in Computer Science, V. Kuncak and A. Rybalchenko, Eds., vol. 7148. Springer, 2012, pp. 396–411. [Online]. Available: https://doi.org/10.1007/978-3-642-27940-9_26

[23] R. Segala and N. A. Lynch, "Probabilistic simulations for probabilistic processes," *Nord. J. Comput.*, vol. 2, no. 2, pp. 250–273, 1995.

[24] M. Shaked and G. Shanthikumar, *Stochastic Orders*, ser. Springer Series in Statistics. Springer, 2007.

[25] A. Sokolova and E. P. de Vink, "Probabilistic automata: System types, parallel composition and comparison," in *Validation of Stochastic Systems - A Guide to Current Research*, ser. Lecture Notes in Computer Science, C. Baier, B. R. Haverkort, H. Hermanns, J. Katoen, and M. Siegle, Eds., vol. 2925. Springer, 2004, pp. 1–43. [Online]. Available: https://doi.org/10.1007/978-3-540-24611-4_1

[26] L. Song, L. Zhang, and J. C. Godskesen, "Bisimulations and logical characterizations on continuous-time Markov decision processes," in *Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, ser. Lecture Notes in Computer Science, K. L. McMillan and X. Rival, Eds., vol. 8318. Springer, 2014, pp. 98–117. [Online]. Available: https://doi.org/10.1007/978-3-642-54013-4_6

[27] L. Zhang, "Decision algorithms for probabilistic simulations," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2009. [Online]. Available: http://scidok.sulb.uni-saarland.de/volltexte/2009/2424/

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-349-5

AALBORG UNIVERSITY PRESS