UNIVERSITÄT
DES
SAARLANDES

Finite Horizon Analysis of Markov Automata

Dissertation zur Erlangung des akademischen Grades des Doktors der Ingenieurwissenschaften (Dr.-Ing.) der Fakultät für Mathematik und Informatik der Universität des Saarlandes

von

Hassan Hatefi Ardakani

Saarbrücken, October 2016

Dean of the faculty	Prof. Dr. Frank-Olaf Schreyer
Day of colloquium	20.12.2016
Chair of the committee	Prof. Dr. Christoph Weidenbach
Reviewers	Prof. Dr. Holger Hermanns Prof. Dr. Peter Buchholz Dr. Sven Schewe
Academic assistant	Dr. Martin Zimmermann

To my parents, my wife and my son

Acknowledgements

This thesis is the result of direct and indirect helps and supports from many people. I greatly appreciate their contributions to this work and would like to say that without their contributions I was not able to successfully accomplish the result of my thesis.

First of all, I would like to thank my supervisor Holger Hermanns for helping me all the way from starting my research to writing this thesis. His liberal view gave me the freedom to explore different directions and develop my research interests. Thank you for giving me the opportunity for meeting and collaborating with different people all around the world.

In addition to Holger, I would like to thank the members of Dependable Systems and Software for their persistence support. Special thanks to Christa Schäfer for her great helps with organisational and bureaucratic procedures. I am grateful for fruitful discussions that I had with my group members, specially with Luis María Ferrer Fioriti. The discussions enabled me to prove the upper bound established in Ch. 4 in addition to many other helpful insights. I also learnt many things from discussions with Yuliya Butkova, Jan Krcál, Vahid Hashemi and Lei Song. Lei, Jan and Vahid, thank you once more for sharing the great experience of teaching quantitative model checking with me. During these years, I am also thankful for having opportunity to work with my colleagues Hernan Baro Graf, Pepijn Crouzen, Christian Eisentraut, Felix Freiberger, Alexander Graf-Brill, Ernst Moritz Hahn, Arnd Hartmanns, Martin Neuhässer, Gilles Nies and Andrea Turini.

This thesis is the result of my collaborations with great people from different research groups other than Dependable Systems and Software. I want to thank my coauthors Florian Arnold, Bernd Becker, Bettina Braitling, Daniel Gebler, Dennis Guck, Joost-Pieter Katoen, Enno Ruijters, Mariëlle Stoelinga and Ralf Wimmer, and tell them it was an invaluable experience for me to collaborate with you during my PhD.

At the end I want to thank with my heart my wife for her love and constant support. She made it possible for me to tackle the tough time I had during writing this thesis after our son was born. Azra, thank you for being with me, supporting me and letting me know what to do when I could not think efficiently!

Abstract

Markov automata constitute an expressive continuous-time compositional modelling formalism, featuring stochastic timing and nondeterministic as well as probabilistic branching, all supported in one model. They span as special cases, the models of discrete and continuous-time Markov chains, as well as interactive Markov chains and probabilistic automata. Moreover, they might be equipped with reward and resource structures in order to be used for analysing quantitative aspects of systems, like performance metrics, energy consumption, repair and maintenance costs. Due to their expressive nature, they serve as semantic backbones of engineering frameworks, control applications and safety critical systems. The Architecture Analysis and Design Language (AADL), Dynamic Fault Trees (DFT) and Generalised Stochastic Petri Nets (GSPN) are just some examples. Their expressiveness thus far prevents them from efficient analysis by stochastic solvers and probabilistic model checkers. A major problem context of this thesis lies in their analysis under some budget constraints, i. e. when only a finite budget of resources can be spent by the model.

We study mathematical foundations of Markov automata since these are essential for the analysis addressed in this thesis. This includes, in particular, understanding their measurability and establishing their probability measure. Furthermore, we address the analysis of Markov automata in the presence of both *reward acquisition* and *resource consumption* within a finite budget of resources. More specifically, we put the problem of computing the *optimal expected resource-bounded reward* in our focus. In our general setting, we support transient, instantaneous and final reward collection as well as transient resource consumption. Our general formulation of the problem encompasses in particular the *optimal time-bound reward and reachability* as well as *resource-bounded reachability*. We develop a sound theory together with a stable approximation scheme with a strict error bound to solve the problem in an efficient way. We report on an implementation of our approach in a supporting tool and also demonstrate its effectiveness and usability over an extensive collection of industrial and academic case studies.

Zusammenfassung

Markov-Automaten bilden einen mächtigen Formalismus zur kompositionellen Modellierung mit kontinuierlicher stochastischer Zeit und nichtdeterministischer sowie probabilistischer Verzweigung, welche alle in einem Modell unterstützt werden. Sie enthalten als Spezialfälle die Modelle diskreter und kontinuierlicher Markov-Ketten sowie interaktive Markov-Ketten und probabilistischer Automaten. Darüber hinaus können sie mit Belohnungs- und Ressourcenstrukturen ausgestattet werden, um quantitative Aspekte von Systemen wie Leistungsfähigkeit, Energieverbrauch, Reparatur- und Wartungskosten zu analysieren. Sie dienen aufgrund ihrer Ausdruckskraft als semantisches Rückgrat von Engineering Frameworks, Steuerungsanwendungen und sicherheitskritischen Systemen. Die Architekturanalyse und Designsprache (AADL), Dynamic Fault Trees (DFT) und Generalized Stochastic Petri Nets (GSPN) sind nur einige Beispiele dafür. Ihre Aussagekraft verhindert jedoch bisher eine effiziente Analyse durch stochastische Löser und probabilistische Modellprüfer. Ein wichtiger Problemzusammenhang dieser Arbeit liegt in ihrer Analyse unter Budgetbeschränkungen, das heisst wenn nur ein begrenztes Budget an Ressourcen vom Modell aufgewendet werden kann.

Wir studieren mathematische Grundlagen von Markov-Automaten, da diese für die in dieser Arbeit angesprochene Analyse von wesentlicher Bedeutung sind. Dazu gehört insbesondere das Verständnis ihrer Messbarkeit und die Festlegung ihrer Wahrscheinlichkeitsmaßes. Darüber hinaus befassen wir uns mit der Analyse von Markov-Automaten in Bezug auf Belohnungserwerb sowie Ressourcenverbrauch innerhalb eines begrenzten Ressourcenbudgets. Genauer gesagt stellen wir das Problem der Berechnung der optimalen erwarteten Ressourcenbegrenzte Belohnung in unserem Fokus. Dieser Fokus umfasst transiente, sofortige und endgültige Belohnungssammlung sowie transienten Ressourcenverbrauch. Unsere allgemeine Formulierung des Problems beinhalet insbesondere die optimale zeitgebundene Belohnung und Erreichbarkeit sowie ressourcenbeschränkte Erreichbarkeit. Wir entwickeln die grundlegende Theorie dazu. Zur effizienten Lösung des Problems entwerfen wir ein stabilen Approximationsschema mit einer strikten Fehlerschranke. Wir berichten über eine Umsetzung unseres Ansatzes in einem Software-Werkzeug und zeigen seine Wirksamkeit und Verwendbarkeit anhand einer umfangreichen Sammlung von industriellen und akademischen Fallstudien.

Contents

Part I

Foundations

Chapter 1

Introduction

The ever-increasing role of computer systems in the modern life pushes up the demand for their safety and reliability. These become essential for safety-critical systems used in medicine, transport, power systems and so on, so as to offer necessary guarantees to assure people and environment are not endangered. Moreover, any computer system is obliged to provide some quality of service in the form of, for instance, energy consumption, throughput, availability, performance . . . . In order to fulfil these requirements, appropriate tools and algorithms for the design and evaluation of computer systems need to be employed.

Today's computer systems are constantly growing in size and complexity. In consequence, the methodology and machinery adopted for their design, evaluation and analysis must (*i*) provide support for the faithful consideration of their important characteristics and properties (*ii*) provide efficient and scalable techniques for their evaluation, and (*iii*) assure that the analysis conducted by these techniques are sound and correct.

In this thesis we utilise *Markov (reward) automata*, an expressive and powerful formalism encompassing a wide range of features that are required for the design and analysis of complex systems. We propose efficient and sound solution techniques for analysis of systems modelled by Markov (reward) automaton. The attention of this thesis is particularly focused upon points (*ii*) and (*iii*), from above, that is to say, upon developing efficient and sound analysis techniques for Markov (reward) automata. Nevertheless, we devote this chapter to point (*i*) by providing necessary motivations for using Markov (reward) automata. In other words, we argue about "why Markov (reward) automata serve as effective modelling formalism for real world systems".

1.1 Modelling real world systems

We elaborate in this section on some of the most common characteristics exhibited by many of the real world systems around us. We clarify the characteristics and the situations in which they appear by giving appropriate examples. We then introduce Markov automaton as the modelling formalism exhibiting those characteristics.

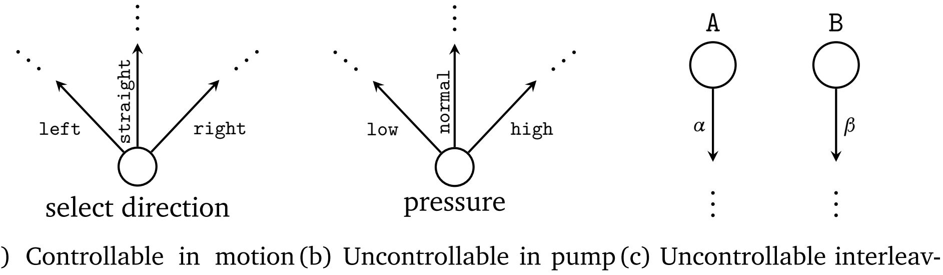
1.1.1 Nondeterminism

Nondeterminism emerges from uncertainty in a system, which can be about its behaviour or its reaction to different internal and external situations. A nondeterministic behaviour is unquantified, that is to say, we cannot quantify how the system behaves upon nondeterminism. Instead we know a collection of possible outcomes of the behaviour. For instance, we do not know what a customer of a vending machine is going to buy. However, we know the collections of items that they can choose from.

We can distinguish between *controllable* and *uncontrollable* nondeterminism. The above is an example of uncontrollable nondeterminism where the choice of the customer can not be done by the vending machine. Typical examples of controllable nondeterminism appear in control systems. For instance, in motion planning of a robot, the controller should decide to go left, right or straight ahead. We can model the possible decisions of the controller as a nondeterministic choice between *left*, *right* and *straight*, as shown in Fig. 1.1a. While modelling, designers may leave the choice open so as to make it possible to later determine the best strategy for the controller.

Uncontrollable nondeterminism often occurs when the system must deal with unclear situations arising by the environment or its own components. The pressure measurements in a pump are completely nondeterministic for a pump controller. The controller may for example be informed of the pressure being in one of three levels *low*, *normal* or *high*, as depicted in Fig. 1.1b. Nevertheless, it does not have any a priori knowledge of the exact pressure level beforehand. Moreover, the delay imposed by the sensors measuring the pressure raises the level of nondeterminism as the controller always acquires a slightly outdated pressure value instead of the current one.

One of the common types of uncontrollable nondeterminism stems from



(a) Controllable in motion planning (b) Uncontrollable in pump control (c) Uncontrollable interleaving

Figure 1.1: Different types of nondeterminism

unobservable behaviour of components of a system that communicate with each other. The exact internal state of the components are usually hidden from outside. It may therefore happen that the order of events triggered by different components is not determined by the system. As an example, illustrated in Fig. 1.1c, two components `A` and `B` run in parallel and *interleave* at the specified states, where they trigger events α and β, respectively. At this point, the events can occur in any possible order. Hence, the possible orders (α before β and vise versa) are modelled by nondeterminism. The situation usually takes place in a system that embodies two or more components running in parallel.

The uncertainty that comes with nondeterminism is unquantified, regardless of being controllable or uncontrollable. In fact, there are other kinds of uncertainty that can be quantified. We can, for instance quantify the behaviour under uncertainty by determining the probability distribution of possible choices. This leads to the concept of probabilistic branching and stochastic timing, which are discussed next.

1.1.2 Probabilistic branching

Probabilistic branching features uncertainty between a countable set of choices. The uncertainty is however quantified and resolved by a probability distribution over the set of possible choices. In other words, the distribution determines the probability of selecting each choice in the uncertainty set. A typical example is tossing a fair coin that has two outcomes, head or tail, each with probability half. Probabilistic branching prevails in different kinds of systems ranging from randomised algorithms and protocols to biological and social systems. We can in general distinguish between two kinds of probability *by design* and probability *by nature*.

By design. Sometimes probabilistic decision making is embedded in the design of algorithms and protocols for specific purposes. *Randomisation* often brings a simple and fast way to solve difficult problems. Computing the minimum cut of a connected graph in a randomised way (Karger's algorithm [Kar93]), determining the delay before retransmission of a frame in Ethernet protocol [Std12], randomised rounding of linear program solutions to integer program solutions (Raghavan-Tompson algorithm [RT87]) and distributed self-stabilising of micro power grids [HH13b] are some examples. Randomisation in such algorithms and protocols is modelled by means of probabilistic branching.

By nature. Some systems may characterise specific natural uncertainty stemming from unpredictable environment. Packets in a wireless network may get lost with a certain probability [GSK05; GSK06] due to the message collision between nodes sending data somewhat at the same time. In social systems for example, the spread of happiness in large social networks [FC08] and the influence between users of such systems [GBL10] are modelled by probabilistic branching. Moreover, discussed by [SB93], Markov models can be used for medical decision making where the health state of a patient upon a clinical event changes probabilistically.

Probabilistic branching reflects phenomena prevailing intentionally or naturally within the structure of a variety of systems. It is therefore an essential tool for modelling and evaluation of real world phenomena. Nevertheless, it is only relevant to the cases when uncertainties are on a set of countable choices. In the next section we deal with a special kind of uncertainty on continuous (uncountable) time domain. That is to say, the uncertainty lies in the time when the next event in a system occurs.

1.1.3 Continuous stochastic timing

Events play a crucial role in explaining the behaviour of different kinds of systems. They are often distinguishable by the time instances at which they occur. This fact is essential for understanding the dynamics of events. As in real systems time evolves continuously, it is natural to study the dynamics of events in continuous domain. Moreover, in many situations, the duration between the occurrences of two events is not deterministic, but rather stochastic. In other words we observe that the events are triggered in random moments. In order to express this behaviour, it is therefore required to model *stochastic timing*.

In theory, the *interarrival* time between two events can be governed by an arbitrary distribution. However, allowing an arbitrary distribution may obscure developing an effective general evaluation technique for systems exhibiting stochastic timing. In particular, it may complicate reasoning in the presence of interaction between their components, e. g. via parallel composition [GBK16]. Moreover, in many applications the interarrival time can be considered to be governed by an *exponential distribution*. Restricting to exponential distributions in one hand simplifies the analysis and on the other hand provides a sound compositional theory [Her02]. Furthermore, an arbitrary general distribution can be represented by a *phase-type distribution* [Cox55]. Since phase-type distributions are constructed by convolution and superposition of several exponential distributions, represented by a Markov chain [Neu75], it is enough to restrict to exponential distributions to produce a general distribution. In addition, efficient solutions for fitting general distributions and measured data to phase-type distributions has been proposed [TBT06]. Moreover, reduction techniques have been developed to minimise the size of the resulting Markov chain [Pul09]. Combining with the compositionality feature, these provide an effective general framework for modelling and analysis of different kinds of systems [HK09; Böd+09].

Regardless of the approximation described above, there are many phenomena that are known to follow exponential distributions. In reliability engineering, exponential distributions are usually adopted for failure-time distributions. In the field of queuing theory, the interarrival time between incoming packets as well as the service time is commonly considered to be exponentially distributed. In biochemical reaction networks, the time to the next reaction to happen is exponentially distributed [Gil76]. From dependability to performance analysis and systems biology, the applications of exponential distributions together with their nice compositionality feature make them one of the powerful tools for modelling and evaluation of real world system.

1.1.4 Markov automata

In the previous sections, we discussed three different behavioural characteristics that are exhibited by a variety of real world systems. We motivates the need for a framework that, not only effectively captures the three features in one model, but also provides automatic generation, reduction and efficient analysis technique for the evaluation of the model.

There are models that individually support each of the three features discussed above. Pure nondeterminism is captured by label transition systems (LTS). Discrete- and continuous-time Markov chains (DTMC,CTMC) support probabilistic branching and stochastic timing, respectively. There have been attempts to combine the features into a single model. In probabilistic automaton (PA) [Seg95], each choice of nondeterminism can branch probabilistically. Interactive Markov chain (IMC) [Her02] orthogonally combines nondeterminism in LTS with stochastic timing in CTMC. None of the mentioned models however exhibit the three features all together.

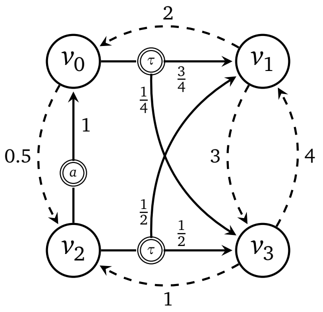
Markov automata (MA) [EHZ10b; EHZ10a] constitute a compositional modelling formalism featuring nondeterminism, probabilistic branching and continuous stochastic timing. They are constructed by combining probabilistic automata with continuous-time Markov chains in an orthogonal way. Therefore, they generalise LTSs, DTMCs, CTMCs, PAs and IMCs. They embody Markov decision processes (MDP)[1] and under certain conditions[2] continuous-time Markov decision processes (CTMDP).

Example 1.1. *An example of a Markov automaton is depicted in Fig. 1.2a. Each of its four states represent a specific internal status of the system that is modelled. Dashed lines, which are annotated with positive rates, illustrate delayed transitions governed by the exponential distributions with the corresponding rates. Solid lines with the small circles in the middle declare transitions with probabilistic branching. The circles are associated with the corresponding events that are triggered while the transitions are performed. Nondeterminism may occur between two probabilistic branches, e. g. in state* v_2.

The model described in Fig. 1.2a cannot be expressed by any of the models mentioned above except for MA. The reason lies in the fact that the model

[1] MDPs can be regarded as a special case of PAs.

[2] Provided that early strategies [NSK09] are considered.



(a) A Markov automaton

	v_1	v_2, a
res.	5	0
t-rew.	2	0
i-rew.	0	1
f-rew.	0.5	0

(b) Its resources and rewards

Figure 1.2: An example of Markov automaton with resource and reward

simultaneously exhibits nondeterminism, probabilistic branching and stochastic timing.

Having an expressive modelling formalism does not necessarily lead to an effective evaluation framework. In fact, the formalism is only useful in practice if it is accompanied with efficient analysis techniques. As an important prerequisite of this, it must be able to model quantities on which the analysis is based. This is discussed in the next section.

1.2 Analysing real world system

An effective analysis of real world systems necessitates modelling their quantitative aspects. They are needed to quantify the behaviour of the system under study. Hence, quantitative aspects are the main tools that provide the designer with a basis on which to decide whether the requirements for the system is satisfied or not. Making such a decision often involves trade-offs: Is the current throughput satisfactory, or is it better to replace the hardware? If so, is that cost- or energy-effective? To answer these questions we need to conduct analyses involving system's resources and services.

1.2.1 Resources and rewards

Essential ingredients of various kinds of analyses are *resources* and *rewards*. Regardless of whether the analysis targets are energy consumption, repair or maintenance costs or other consumable quantities, it involves quantifying a resource consumption of some kind. Furthermore, throughput, availability, safety and many other services that are provided by systems can be formulated via the reward gained by the system. In this section, we explain resources and rewards separately in our setting, we then draw the distinction between them, and discuss why they are useful and effective.

Resources. *Time* is the principal resource that, by advancing, is consumed in both discrete- or continuous- time models. In many cases however, considering time as the only resource is restrictive. There are applications that concern energy consumption, repair or maintenance costs as the main resources. We therefore generalise the concept of time to *resources*. In this view, *time*, *energy*, *money* or any consumable quantities spent by a system can be considered as resources. Resources are usually either *limited* in quantity or they need to be carefully *budgeted*. In both cases, designers might be interested in analysing some properties in their models under a given budget of resources.

Rewards. Rewards are used to quantify the services that are offered by a system. They can for instance quantify the *availability* of a system by measuring the percentage of time a system is ready to provide services. In this example, rewards are obtained in a *time-dependent* manner. Rewards can also be *event-dependent* as in *throughput*, which asses the rate of task completions in a system. In this case, rewards are granted upon event occurrences. Rewards may as well depend on *the state of a system*. A typical example is *system safety*, which evaluates the probability of reaching states at which the system is safe within a specified period of time. Rewards are often associated with the result produced or the service offered by a system. Hence, the reward acquisition can be directly influenced by nondeterministic and stochastic behaviour of a system.

The distinction. There are some quantities that can be considered as both resources and rewards. For instance, time in availability servers as a reward quantity whereas in safety it is a resource. There is however a subtle difference between resources and rewards. Resources are always limited quantities, or restrictions in the form of fixed budgets are imposed on their consumption. They are usually related to the quantities that are consumed by a system. Rewards, on the other hands are related to the quantities that are produced or offered by a system. They are obtained without concerning about any budget. In availability, time measures the quality of being ready to offer services. In safety however, a deadline is set on the time span of the system.

Why resources and rewards. It is possible to consider, depending on our view, a quantity as rewards or resources. Being able to model resources and rewards is crucial when analysing real world systems. First of all, analyses not involving resource consumption and reward acquisition are very limited and cannot indeed answer important questions regarding the functionality and the performance of the systems under study. Resources and rewards on the other hand, can be deployed with a considerable degree of flexibility to capture a wide range of system's measurements. Hence, they open up an opportunity to model various kinds of quantitative aspects of real world systems. Moreover, the abstract view of various quantities as resources and rewards generalises the way analyses can be specified and carried out. For instance, if there is an analysis technique for maximum rewards gained within a deadline, it can be deployed for both computing the maximal throughput and the maximal system safety within the deadline, depending on how the rewards are defined. Altogether, resources and rewards form the basis for effective system analysis in a single framework.

In the next section we equip Markov automata with resources and rewards.

1.2.2 Markov reward automata

Markov reward automata (MRA) are obtained by incorporating resource and reward structures into MAs. They enjoy the expressiveness of MAs combined with the ability to capture a wide range of useful quantities that can be expressed by resources and rewards. This makes them an ideal framework for modelling and analysis of real world systems. In the sequel, we briefly discuss how to boost MAs with resource and reward.

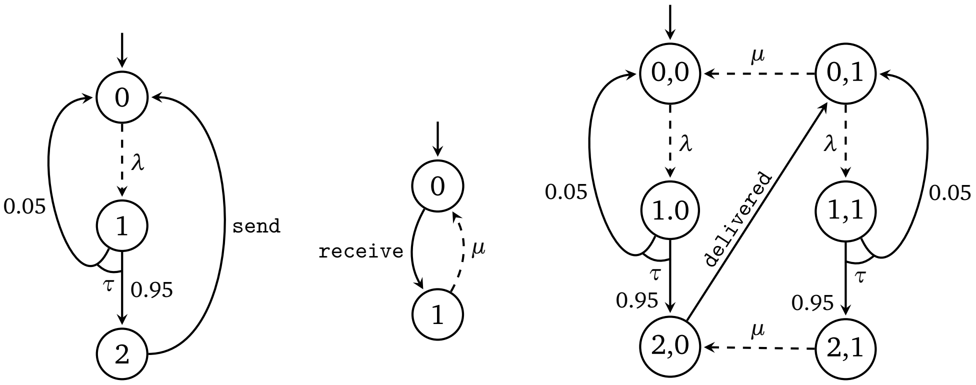
In this thesis, we assume resources are spent with a non-negative constant rate while the system stays at a specific state. The rates may indeed vary from one state to another. The rate may also be zero at some state, meaning that the system has no resource consumption at the state. For example, as described in Tab. 1.2b, the resource is spend in state v_1 of the MA shown in Fig. 1.2a with rate 5. This means that the MA consumes 5 units of resource if it stays in v_1 for one time unit. Similarly to the resources, *transient rewards* increase linearly with time. The exemplary MA gains 2 units of transient reward per time unit of staying at v_1. MRAs are able to capture rewards that are time independent. They can be granted either *instantaneously* upon triggering an event, i. e. executing a transition, or *finally* when the specified budget of resources is empty. In our example, one unit of instantaneous reward is obtained by executing a from v_2. Furthermore, half a unit of final reward is gained when the MA runs out of the resource budget while visiting v_1.

In the next section we compile a rich list of the useful features of MRAs that are discussed above. With this we can strongly argue why MRAs are perfect tools for quantitative system analysis.

1.3 Why Markov reward automata

Here we summarise all the benefits of employing MRAs as the modelling formalism for quantitative system analysis.

Expressiveness and generality. We explained in Sec. 1.1 that MAs, as the basic block of MRAs, support useful features in three orthogonal dimensions: nondeterminism, probabilistic branching and continuous stochastic timing. It makes MRAs expressive enough to capture behavioural aspects of a diverse range of systems that are used in practice. Moreover, MRAs generalise the behavioural



(a) The producer (b) The consumer (c) Their parallel composition

Figure 1.3: A producer-consumer model in MA

models that partially support these dimensions, among all: LTS, DTMC, CTMC, PA and IMC. This is practically interesting since their application domains are carried over to the MRA world and also theoretically important as properties established for MRAs directly hold for the sub-models.

Compositionality. Compositionality is a key feature that facilitates modular modelling of highly complex systems. It enables designers to divide (or assemble) a system into (or out of) several subsystems, each called a *component*, in a top-down (or bottom-up) manner. The dynamics of the system is then defined via composing its components in *parallel*. The components may need to communicate with each other to send and receive information. Compositionality for MRAs is achieved by *parallel composition*. Parallel composition prescribes the way the components of a system interact via *synchronisation* through *actions*. Informally speaking, it determines how two components can synchronise their executions by performing a shared action. This abstract view can, for instance be instantiated as one component sending an item to another one. MAs are compositional and so are MRAs. It is thus possible to model a complex system by MRAs via, firstly breaking its structure down into several interacting components, secondly modelling each component by an MRA, and lastly putting the component in parallel composition via a set of appropriate actions.

Example 1.2. *Fig. 1.3 shows Markov automaton models of a producer (Fig. 1.3a), a consumer (Fig. 1.3b) and their parallel composition (Fig. 1.3). The states of the composed model are formed by pairing the corresponding states of its components. The producer and the consumer both start at state* 0. *The producer produces an item in a randomly exponential duration with average* $\frac{1}{\lambda}$ *time units. The item may however be corrupted with a probability of* 0.05. *The proper item is sent over to the consumer through synchronisation between actions* `send` *and* `receive`. *The* *synchronisation takes place at state* $(2,0)$ *and as a result, the item is* `delivered` *to the consumer. The item is then consumed in a random amount of time that is exponentially distributed with rate* μ*. Action* τ *represent an internal behaviour that is invisible from outside.*

Compositionality brings a number of important advantageous. First of all, it perfectly matches the design procedures that are commonly used in practice when developing systems. Either of top-down or bottom-up approach splits a system into some interacting components, which is exactly the procedure supported by compositional modelling. Secondly, it provides the basis for automatic model construction specially for complex and huge systems. Design reusability, ease of debugging and effective minimisation are some other benefits offered by compositionality.

MRA as a semantical model. Expressiveness and generality of MRAs enables them to serve as the semantic foundation of, among all, stochastic activity network (SAN) [MMS85], generalised stochastic Petri nets (GSPN) [Eis+13], dynamic fault trees (DFT) [BCS10], Statecharts [Böd+09] and AADL, the architecture analysis and design language [Boz+11]. Hence their application domains are inherited by MRAs. It is therefore worthwhile to study MRA not only because of their considerable potential for modelling of real world systems directly, but also since they deliver the low level semantics of the mentioned high level frameworks.

Efficient modelling. An efficient modelling methodology is essential for any formalism, specially to be used for describing extensive and complicated systems. It is neither appropriate nor efficient to directly specify a model at a low level of abstraction, e. g. listing states and transitions. Therefore, formal approaches usually provide a high level language for model specification. The model is then generated from a high-level description using automatic tools. Without this it is hardly possible to specify huge and complex systems.

An efficient modelling approach exists for MRAs. Markov automata process algebra (MAPA) [TPS13; Tim13] serves as the high-level language for describing MRAs. In addition, various kinds of reduction techniques are proposed [Tim13] that can reduce the size of the model while preserving a large class of properties, including the ones we focus in this thesis.

Efficient analysis. Without support for analysis, any formal model provides very limited insight into the modelled system. It is therefore necessary for formal frameworks to support useful and practical analyses. A wide range of analyses has been proposed for MRAs. Efficient algorithms have been successfully developed for time- and resource-bounded reachability, unbounded reachability, expected goal-bounded rewards, long-run average rewards and expected time-

and resource-bounded rewards. This range helps designers to quantitatively evaluate their systems in different aspects.

Tool support. The success of MRA as a modelling and analysis formalism is only guaranteed if it is accompanied by supporting tools. The modelling and analysis techniques discussed above, all have been implemented in MAMA tool chain [Guc+14b; Guc+14a]. This successfully integrates two separate tools for the modelling and for the analysis of MRAs. For the modelling, SCOOP [Tim11] provides an efficient generation of MRAs from MAPA specifications, accompanied by the reduction techniques mentioned above. In the analysis part, IMCA [Guc12] supports all the aforementioned analyses. We explain in the next section which of those analysis are consider in this thesis and why they are important.

1.4 Why resource-bounded rewards

Finite vs infinite horizon. The title of this thesis puts emphasis on *finite horizon* as the overwhelming type of analyses done for MRAs. Being on finite horizon means that the analyses imposes a restriction on the amount of resources that can be spend. This restriction is given in the form of a *resource budget*. On the contrary, analyses on *infinite horizon* deal with objectives with the unbounded resource budget, i. e. there is no restriction on the resource consumption. In comparison with infinite horizon, analyses on finite horizon are usually more difficult to be carried out. This is due to the fact that, for the analyses on finite horizon, we may need to change nondeterministic decisions at arbitrary time points [Mil68; MMS85; BS11; RS11]. By contrast, on infinite horizon the decisions are static along the time line [Put05; Guc+14a; Guc+14b].

The optimal ERR. Being on finite horizon in our setting coincides with resource boundedness. Here, resource consumption and reward acquisition are intertwined. More precisely, rewards are obtained while resources are consumed by MRAs. We are interested in computing *resource-bounded rewards*, the rewards that are obtained up to running over a given resource budget. However, nondeterministic and stochastic behaviour of MRAs has direct impact on their reward acquisition. The nondeterministic choices that are made and the stochastic dynamics of the model govern the reward acquisition. Therefore, the amount of rewards obtained by an MRA is both nondeterministic and stochastic. After fixing all nondeterministic decisions, the resource-bounded reward acquisition is stochastic and forms a *random variable*. The random variable is thereby governed by the nondeterministic decisions that are made in the model. The expectation of the random variable yields the *expected resource-bounded reward*. This refers to a range of expectations, each corresponding to a specific resolution of nondeterministic choices in the model. The *minimum* and the *maximum* in this range are of special interest. These help us to measure the worst and the best services that a model can offered. To this end, we aim to compute the minimal or the maximal expected resource-bounded rewards (ERR) for MRAs. More specifically, a finite budget of resource is given, the model then tries to gain as much as (or as little as) rewards possible in expectation until running over the budget. Here we clarify why computing the optimal ERR is important.

Naturalness: It is completely natural to allocate a limited budget for the amount of resources that can be consumed in a system. We often face in different systems, a limited life time, a fixed planned budget for repair and maintenance costs or a battery with a limited capacity. The analyses in all of these cases necessitate the resource consumption to be within certain bounds.

Expressiveness: Due to the flexibility of deploying resources and rewards in a model, it is possible to conduct a diverse range of analyses in the framework of resource-bounded rewards. The optimal time-bounded rewards, time- and resource-bounded reachability are some examples.

Challenge: Computing resource-bounded properties represent significant theoretical and technical challenges. The analyses on infinite horizon (unbounded budget) on the other hand, are usually equivalent to problems in the same class but in discrete domain, which are well-understood and relatively easy to solve.

Achieving the potential of MRAs: Using MRAs only for infinite horizon analyses neglects their main strength as a continuous time model. Most of the analyses can be done on discrete-time models without need to introduce a new extensive and expressive model in continuous time. We strongly believe that resource-bounded analyses helps to achieve the full potential of MRAs as a continuous stochastic-time model.

In addition, studying analyses on finite horizon properties helps us to better understand the infinite horizon ones since the latter can be seen as the limit of the former. For instance, the long-run average reward is the limit of time-bounded reward divided by the time bound as time bound goes to infinity. These points urge us to consider the optimal ERR as the main property that we study on MRAs.

1.5 Contributions

This thesis contributes to the field of quantitative system analysis. We look at MRAs and study their analysis on finite horizon. More specifically, we aim to establish a sound theory and efficient techniques for computing the optimal expected resource-bounded reward (ERR). The analysis is considered as both challenging and useful in practice. It is in general much more difficult than the similar problems on infinite horizon. We develop all the way from a solid foundation in theoretical basics of MRAs to characterisations of the optimal ERR. We then introduce an approximation scheme based on discretisation with proved error bound for conducting the analysis. This leads to a numerically stable polynomial time algorithm for computing the optimal ERR. We summarise our contributions in five dimensions.

- We establish a solid theoretical foundation for MRAs. This includes in particular, definition of meaningful strategies and a unique probability measure for MRAs. The uniqueness ensures that with the predefined semantics, there is one and only one valid way to analyse MRAs. Accordingly, we propose a useful and solid way to define the optimal ERR via sequences of functions. And lastly as a fundamental result, we introduce *expectation splitting* on MRAs, which is, informally speaking, expressing expectation on infinite executions in terms of expectation on finite executions. This forms a crucial basis for several contributions throughout this thesis.

- We thoughtfully classify MRAs according to the finiteness of their optimal ERR. As a result, we distinguish between *reward convergent* and *reward divergent* models. A reward divergent model can infinitely obtain rewards by spending only a finite amount of resources. This generalises the concept of *timelock*, i. e. taking infinitely many transitions in a finite duration. At the end, we prove a crucial upper bound on the optimal ERR of reward divergent models.

- We study in depth the characterisations of the optimal ERR. First and foremost, we introduce its characterisation as a fixed point of a Volterra integral equation. With this, we show for the class of reward convergent models that the optimal ERR is Lipschitz continuous with respect to resource bound. We provide the corresponding Lipschitz constant. We also show that the optimal ERR may not be differentiable at the points a nondeterministic choice has to be changed.

- We develop a sound and stable technique for computing the optimal ERR. We first consider the optimal expected time-bounded rewards (ETR) as a special case of the optimal ERR where the role of resources is taken by time. To compute this approximately, we propose a sound discretisation scheme with strict error bound that leads to a numerically stable polynomial time algorithm. To employ the algorithm for the general case of ERR, we introduce a resource-to-time measure preserving transformation by *rate scaling* that reduces the resources-bounded to time-bounded computation.

- We demonstrate the effectiveness of our analysis techniques in practice. We have implemented the tools and algorithms for computing the optimal ERR. Moreover, we have assembled an extensive collection of industrial and academic case studies. The empirical evaluation of our computational technique in various dimensions shows that it is useful in practice and also brings value into quantitative analysis of real world system.

1.6 Overview

The rest of the thesis is organised as follows.

Chapter 2 provides the necessary background mostly in mathematical analysis and probability theory that is used in this thesis.

Chapter 3 defines the Markov automaton model, its semantics and its probability measurability. This includes proposing measurable strategies, and expectation splitting.

Chapter 4 studies the optimal ERR in detail by looking into its measurability and finiteness. The classification of MRAs into reward convergent and reward divergent models with an upper bound for the former is also included in this chapter.

Chapter 5 characterises the optimal ERR as a fixed point of a Volterra integral equation and then inspects its Lipschitz continuity and its differentiability.

Chapter 6 develops an approximation scheme for computing the optimal ETR. In this chapter, a measure preserving transformation is introduced that reduces the computation of the optimal ERR to ETR.

Chapter 7 presents an empirical evaluation of the computational technique proposed in this thesis over an extensive collection of case studies.

Chapter 8 concludes the thesis by summarising the main contributions of this thesis. It also discusses the current restrictions and open questions with a list of possible future works.

The beginning of each chapter provides motivations for and overview of the content. The end of each chapter summarises the results and the contributions, overviews the related works and discusses open questions and future extensions.

Chapter 2

Mathematical Background

This chapter provides the mathematical background that is necessary for understanding the concepts, methods and proofs used in the thesis. We starts with the basic notations for sets and functions and then look into measure theory. Afterwards we discuss some concepts in mathematical analysis including sequences of functions, their convergence and also continuity. At the end some useful tools in probability theory are represented.

2.1 Basic notations

We introduce here the notations to be used throughout this thesis.

Sets. We use operator "$\setminus$" for set difference, "$\bigcup_i$" and "$\bigcap_i$" for the union and the intersection of a family of sets indexed by i. The *empty set* is written as $\emptyset$. The set of all subsets of set A is denoted by 2^A. The set of real and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$. We extend real numbers with $\mp\infty$ to obtain the set of *extended real numbers*, denoted by $\overline{\mathbb{R}}$. The set of *extended natural numbers* is similarly defined by $\overline{\mathbb{N}} := \mathbb{N} \cup \{\infty\}$. The set of (extended) *nonnegative* and respectively *positive* reals numbers are denoted by $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$ ($\overline{\mathbb{R}}_{\geq 0}$ and $\overline{\mathbb{R}}_{>0}$).

Functions. We write $f : A \to B$ to denote function f from X to Y. The *preimage* of B under f is denoted by $f^{-1}(B) := \{x \in X \mid f(x) \in B\}$. If a partial order is given for Y, say $(Y, \leq)$, $\{f \leq y\}$ is an abbreviation for $\{x \in X \mid f(x) \leq y\}$. For $f, g : X \to Y$, $f \leq g$ states f is pointwise less than g, i. e. $f(x) \leq g(x)$ for every $x \in X$. For function f over $X \times Y$, $f(x, \cdot)$ with $x \in X$ refers to the mapping $y \to f(x, y)$ for $y \in Y$. The *indicator* of $X \subseteq U$ is the mapping $\mathbb{1}_X : U \to \{0, 1\}$ with $\mathbb{1}_X(x) := 1$ if $x \in X$, and $\mathbb{1}_X(x) := 0$ if $x \notin X$. The *zero* function, which maps every thing to zero, is denoted by $\mathbf{0}$.

2.2 Measure theory

Measure theory provides an abstract approach for measuring the size of sets. Size in this context may refer to length, area, volume or, in particular probability. This section presents the necessary concepts about measures and also probability measures that is required for understanding the theory developed in this thesis.

2.2.1 σ-algebras and measures

In general a measure is a function, say μ, that assigns the size of A, $\mu(A)$, to every set A in a specific universe. A collection of the subsets of the universe for which the size exits is referred to as *σ-algebra*. It exhibits some intuitive properties, for instance, if the size of A_1 and A_2 exist, then their union size must exist as well. Formally, σ-algebra is closed under set complement, countable union and countable intersection.

Definition 2.1 (σ-algebra)**.** *A collection $\mathcal{X}$ of subsets of $\mathbb{X}$ ($\mathcal{X} \subseteq 2^{\mathbb{X}}$) is a σ-algebra iff it satisfies the following conditions:*

(a) $\mathbb{X} \in \mathcal{X}$,

(b) *if* $A \in \mathcal{X}$, *then* $(\mathbb{X} \setminus A) \in \mathcal{X}$,

(c) *if* $A_1, A_2, \ldots \in \mathcal{X}$, *then* $\bigcup_{i=1}^{\infty} A_i \in \mathcal{X}$.

Accordingly, the pair $(\mathbb{X}, \mathcal{X})$ *is referred to as* measurable space.

The closure under countable intersection follows from (b) and (c).

Example 2.2. *Let $\mathbb{X} = \{1, 2, \ldots, 6\}$ correspond to the set of outcomes of throwing a six-sided dice. The power set of $\mathbb{X}$, $2^{\mathbb{X}}$, satisfies all conditions of Def. 2.1 and is thereby a σ-algebra. The same for instance holds for*

$$\big\{\emptyset, \{1\}, \{2\}, \{1,2\}, \{3,4,5,6\}, \{2,3,4,5,6\}, \{1,3,4,5,6\}, \mathbb{X}\big\}$$

A *measurable set* refers to an element of a σ-algebra. The size of a measurable set is given by a *measure*, which is a *set function* that assigns a nonnegative number to a given set in a specific universe. It is formally defined as follows.

Definition 2.3 (Measure)**.** *A* measure *on a σ-algebra $\mathcal{X}$ of set $\mathbb{X}$ is the mapping $\mu : \mathcal{X} \to \overline{\mathbb{R}}$ such that*

(a) $\mu(A) \geq 0$ *for* $A \in \mathcal{X}$,

(b) *it is* countably additive*, that is for pairwise disjoint sets $A_1, A_2, \ldots \in \mathcal{X}$, it holds that*

$$\mu\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mu(A_i)$$

Accordingly, the triple $(\mathbb{X}, \mathcal{X}, \mu)$ *is referred to as* measure space. *Moreover, if* $\mu(\mathbb{X}) = 1$*, then* μ *and* $(\mathbb{X}, \mathcal{X}, \mu)$ *are called* probability measure *and* probability space*, respectively.*

Specially in the context of probability, $\mathbb{X}$ refers to as *sample space*. It represents the set of all possible outcomes of a random experiment. Then each elements of $\mathbb{X}$ corresponds to a certain outcome of the random experiment. Moreover, an element of σ-algebra is also called a *measurable event* or simply an *event*.

Events can have complex structures. They can also be combined via set operations, making them more and more complicated. However, there are usually "basic events" corresponding to the elementary outcomes of a random experiment. In rolling a dice for example, each face can be seen as a basic event. More complex events are then build upon the basic events using set operations.

It is usually enough to know basic events that can exist in a sample space (or other kinds of spaces) in order to construct a σ-algebra containing all possible events. This is done by constructing the closure under set operations described in Def. 2.1. Let E be the set of some basic events, then $\underaccent{\sim}{\sigma}^\star(E)$ denotes the *smallest* σ*-algebra* generated by the set of events in E. Moreover, it is possible to extend the measure that intuitively speaking, is defined only on the basic events E to the measure on $\underaccent{\sim}{\sigma}^\star(E)$ in a unique way. We will make use of this construction later in this chapter and also in the following chapters.

2.2.2 Measurable functions

The concept of measurability can be extended from sets to functions. *Measurable functions* exhibit useful properties. In particular, the theory of *Lebesgue integration* applies to the class of functions that enjoy a certain measurability feature.

Definition 2.4 (Measurable function). *Let* $(\mathbb{X}_i, \mathcal{X}_i)$*,* $i = 1, 2$ *be two measurable spaces. Function* $f : \mathbb{X}_1 \to \mathbb{X}_2$ *is* measurable *relative to* σ*-algebras* $\mathcal{X}_1$ *and* $\mathcal{X}_2$*, also written as* $f : (\mathbb{X}_1, \mathcal{X}_1) \to (\mathbb{X}_2, \mathcal{X}_2)$ *iff* $f^{-1}(A) \in \mathcal{X}_1$ *for every* $A \in \mathcal{X}_2$*.*

The preimage of any measurable set under a measurable function is again measurable. This means that the inverse image of a measurable function preserves measurability. We do not need to check this preservation for every measurable set. Indeed, it is sufficient to check it only for the basic events (see e. g. [AD99, Definitions and Comments 1.5.1]).

In Lebesgue integration process, *Borel measurable functions* play an important role. They belong to a certain class of measurable functions. To explain this class we first need to define *Borel* σ*-algebra*. It is the σ-algebra over the set of real numbers generated by the set of open intervals.

Definition 2.5. *Borel* σ*-algebra on* $\mathbb{R}$*, denoted by* $\mathcal{B}(\mathbb{R})$*, is the smallest* σ*-algebra generated by open intervals, i. e.* $\mathcal{B}(\mathbb{R}) := \underaccent{\sim}{\sigma}^\star(\{(a, b) \mid a, b \in \mathbb{R}\})$*. An element of Borel* σ*-algebra is called* Borel set*.*

Some observations can be made on the Borel σ-algebra. First of all, it can also be constructed by other types of intervals as the basic events, for instance by closed, half closed interval, or interval of the form $(-\infty, b]$ for $b \in \mathbb{R}$. Secondly, Borel σ-algebra can be similarly defined on a segment of reals. In this way, we can for example construct $\mathcal{B}(\mathbb{R}_{\geq 0})$ and $\mathcal{B}([0,1])$. The σ-algebra can also be defined on the extended reals. Borel sets in this σ-algebra are the elements of $\mathcal{B}(\mathbb{R})$, but in addition they may contain any subset of $\{-\infty, +\infty\}$. With this observations, we proceed by the definition of Borel σ-algebra.

Definition 2.6 (Borel measurable function). *Function $f : (\mathbb{X}_1, \mathcal{X}_1) \to (\mathbb{X}_2, \mathcal{X}_2)$ is* Borel measurable *iff $\mathcal{X}_2$ is a Borel σ-algebra.*

As mentioned before, it is enough to show the measurability of a function for basic events. In other words, a function is measurable if and only is its inverse image under every basic event is measurable. This is specially the case for Borel measurable functions.

Proposition 2.7. *Let $(\mathbb{X}, \mathcal{X})$ be a measurable space. Function $f : \mathbb{X} \to \overline{\mathbb{R}}$ is Borel measurable iff $\{x \in \mathbb{X} \mid f(x) \leq b\} \in \mathcal{X}$ for all $b \in \mathbb{R}$.*

Proof. It directly follows from the fact that the Borel σ-algebra can be constructed by $(-\infty, b]$ for $b \in \mathbb{R}$. □

Borel measurable functions have certain features that make them useful in mathematical analysis. Among them is their closure under arithmetic operations.

Proposition 2.8. *For $f_1, f_2 : (\mathbb{X}, \mathcal{X}) \to (\overline{\mathbb{R}}, \mathcal{B}(\overline{\mathbb{R}}))$ it holds that $f_1 + f_2$, $f_1 \cdot f_2$, f_1/f_2 and $k \cdot f_1$ for every $k \in \mathbb{R}$ are Borel measurable providing the functions are well defined.*

Proof. See the proof of [AD99, Thm. 1.5.6]. □

In this context, arithmetic operations on functions are well defined when they do not lead to $\infty - \infty$, $a/0$ and ∞/∞.

Another important feature of Borel measurable functions is that they are closed under limit sequence. In other words, the limit of a sequence of Borel measurable functions is Borel measurable.

Theorem 2.9. *Let $f_n : (\mathbb{X}, \mathcal{X}) \to (\overline{\mathbb{R}}, \mathcal{B}(\overline{\mathbb{R}}))$ for $n \in \mathbb{N}$ be a sequence Borel measurable functions that converge pointwise to $f : \mathbb{X} \to \overline{\mathbb{R}}$, i. e. $\lim_{n \to \infty} f_n(x) = f(x)$ for all $x \in \mathbb{X}$. It then holds that f is Borel measurable.*

Proof. See the proof of [AD99, p. 1.5.4]. □

Sequences of functions will be discussed in detail in Sec. 2.4.2.

2.2.3 Extension of measures

Assume two measure spaces are given; here we explain how to build a measure space upon them. This can be seen as the generalisation of Cartesian products for measure spaces. In particular, we can use this technique for modular construction of measure and probability spaces.

We start with the construction of σ-algebras over the Cartesian product of two sets, which is known as *product σ-algebra*. For this, we first introduce *measurable rectangles*, which are obtained by Cartesian product of two measurable sets.

Definition 2.10 (Measurable rectangle)**.** *Given two measurable spaces* $(\mathbb{X}_1, \mathcal{X}_1)$ *and* $(\mathbb{X}_2, \mathcal{X}_2)$ *then* $A_1 \times A_2$ *is a* measurable rectangle *iff* $A_1 \in \mathbb{X}_1$ *and* $A_2 \in \mathbb{X}_2$*.*

Measurable rectangles are the basic blocks of the product σ-algebra. In other words, the product σ-algebra is generated by the set of all measurable rectangles.

Definition 2.11 (Product σ-algebra)**.** *The* product space *of two measurable spaces* $(\mathbb{X}_1, \mathcal{X}_1)$ *and* $(\mathbb{X}_2, \mathcal{X}_2)$ *is defined as* $(\mathbb{X}_1 \times \mathbb{X}_2, \mathcal{X}_1 \otimes \mathcal{X}_2)$*, where* $\mathcal{X}_1 \otimes \mathcal{X}_2$ *is the* product σ-algebra*, which is the smallest σ-algebra that contains all measurable rectangles, i. e.*

$$\mathcal{X}_1 \otimes \mathcal{X}_2 = \underline{\sigma}^{\star}\big(\{A_1 \times A_2 \mid A_1 \in \mathcal{X}_1, A_2 \in \mathcal{X}_2\}\big)$$

We can now construct a measure on the product σ-algebra. The construction assures that such a measure exists and it is also unique. In our setting, we heavily utilise this construction to establish the probability measure for MRAs. Intuitively speaking, if we have the probability measure for one step execution of MRAs, we can construct the probability measure on the product of n steps. This technique makes use of *product measure theorem*. We present the theorem in probabilistic setting. For the theorem in more general setting see for instance [AD99, Thm. 2.6.2].

Theorem 2.12 (Product measure theorem)**.** *Let* $(\mathbb{X}_1, \mathcal{X}_1, \mathfrak{p}_1)$ *be a probability space and* $(\mathbb{X}_2, \mathcal{X}_2)$ *be a measurable space. Assume that for every* $x_1 \in \mathbb{X}_1$ *we have a probability measure* $\mathfrak{p}_2(x_1, \cdot) : \mathcal{X}_2 \to [0,1]$ *with* $x_1 \to \mathfrak{p}_2(x_1, A)$ *being Borel measurable for every* $A \in \mathcal{X}_2$*. Then there exist a* unique *probability measure* $\mathfrak{p}$ *on* $\mathcal{X}_1 \otimes \mathcal{X}_2$ *such that*

$$\mathfrak{p}(A) = \int_{\mathbb{X}_1} \mathfrak{p}_2(x_1, A(x_1))\, \mathfrak{p}_1(\mathrm{d}x_1) \qquad \text{for } A \in \mathcal{X}_1 \otimes \mathcal{X}_2$$

where $A(x_1) = \{x_2 \in \mathbb{X}_2 \mid (x_1, x_2) \in A\}$*.*

Proof. See the proof of [AD99, Thm. 2.6.2]. □

We usually refer to probability measures $\mathfrak{p}_1$ and $\mathfrak{p}_2$ as “sub-measures”. Once they are known and satisfy the condition of the product measure theorem, we can combine them to constitute the probability measure on the product space in a unique way. The crucial step is of course to show that they both satisfy the conditions of the product measure theorem. The result of the product measure theorem can be extended in order to compute the expectation of Borel measurable functions on a product space. That is to say, if we have a Borel measurable function on the produce space, it is possible to compute its expectation using the sub-measures. This is known as Fubini’s theorem.

Theorem 2.13 (Fubini’s theorem). *Assume the hypothesis of the product measure theorem* (Thm. 2.12) *and let* $f : \mathbb{X}_1 \times \mathbb{X}_2 \to \overline{\mathbb{R}}_{\geq 0}$ *be a Borel measurable function. Then,*

$$\int_{\mathbb{X}_1 \times \mathbb{X}_2} f \, \mathrm{d}\mathfrak{p} = \int_{\mathbb{X}_1} \int_{\mathbb{X}_2} f(x_1, x_2)\, \mathfrak{p}_2(x_1, \mathrm{d}x_2)\, \mathfrak{p}_1(\mathrm{d}x_1)$$

Proof. See the proof of [AD99, Thm. 2.6.4]. □

The product measure theorem and Fubini’s theorem can be seen as the tools for “integration” of probability measures and expectation computation. They can on the other hand be viewed as “disintegration” tools. We utilise this feature in particular in Ch. 3 for *expectation splitting*, which is an essential technique for computing expectations in MRAs.

2.3 Probability theory

This section provides background concepts of probability from a measure theoretic points of view. It includes the concept of distributions, random variables and expectations.

Distribution. Let X be a finite (or countably infinite) set. A probability distribution over X is a function $\mu : X \to [0, 1]$ such that $\sum_{x \in X} \mu(x) = 1$. The *support* of μ, denoted by $\mathrm{supp}(\mu)$, is the set of all elements of X that gives positive probability, i. e. $\mathrm{supp}(\mu) := \{x \in X \mid \mu(x) > 0\}$. A distribution is *Dirac* on $x \in X$, written as Δ_x, iff $\Delta_x(x) = 1$. It is not hard to see that a distribution induces a probability measure on the power set of X. The set of all probability distributions over X is denoted by $\mathrm{Distr}(X)$.

Random variable. In a probability space $(\mathbb{X}, \mathcal{X}, \mathfrak{p})$, $\mathbb{X}$ represents the set of possible outcomes, and thereby and element of $\mathbb{X}$ corresponds to a random experiment. However, it may not be relevant to directly work with $\mathbb{X}$ since it can have an arbitrary shape and complex structure. Instead, we use a mechanism that measures a certain quantity of random experiments. This is usually a function X that maps outcome x to (extended) real number $X(x)$. The inverse image of such a function forms a set of outcomes that must be indeed measurable. Thus a random variable is required to be Borel measurable.

Definition 2.14 (Random variable)**.** *An* (extended real-valued) random variable *X on probability space $(\mathbb{X}, \mathcal{X}, \mathfrak{p})$ is a Borel measurable function from $\mathbb{X}$ to $\overline{\mathbb{R}}$.*

As the definition suggests the terms "random variable" and "Borel measurable function" are equivalent in probability domain.

Expectation. Borel measurability of random variables makes further analysis possible. In particular we can compute their expectation, which is intuitively their average value. Expectation provides a clue on how the random experiments are quantified on average according to their likelihood.

Definition 2.15 (Expectation)**.** *The* expectation *of random variable X on $(\mathbb{X}, \mathcal{X}, \mathfrak{p})$ is defined as*

$$\mathbb{E}(X) := \int_{\mathbb{X}} X \, \mathrm{d}\mathfrak{p}$$

We finish this section by looking into the expectation of a special random variable, namely the one that maps each random experiment to a natural number. In our setting, such a random variable is used for instance to count certain events during the execution of an MRA.

Lemma 2.16. *Let X be a* nonnegative integer-valued random variable *on $(\Omega, \mathcal{F}, P)$, then*

$$\mathbb{E}(X) = \sum_{i=0}^{\infty} P(X > i)$$

Proof. Consider the sum:

$$\begin{array}{rcccccl} S = P(X=1) & + & P(X=2) & + & P(X=3) & + \cdots \\ & + & P(X=2) & + & P(X=3) & + \cdots \\ & & & + & P(X=3) & + \cdots \\ & & & & & + \cdots \end{array}$$

The i-th column elements sum up to $i \cdot P(X=i)$, then $S = \sum_{i=0}^{\infty} i \cdot P(X=i) = \mathbb{E}(X)$. On the other hand, the i-th row elements sum up to $\sum_{j=i}^{\infty} P(X=j) = P(X \geq i)$. Hence $\mathbb{E}(X) = \sum_{i=1}^{\infty} P(X \geq i) = \sum_{i=0}^{\infty} P(X > i)$. □

2.4 Analysis

This chapter includes several definitions and results in mathematical analysis that are crucial for understanding of the material in this thesis. We first look into Lipschitz continuity and then discuss sequences of functions and their convergence. We then give arguments in different cases how we can interchange between limiting operators regarding a sequence of functions.

2.4.1 Lipschitz continuity

There are different notions of continuity of real-valued functions. Lipschitz is a strong notion of continuity asserting that the variation of a function is bounded. In other words, the function cannot vary faster than some finite rate. Here we provide the definition for real-valued functions.

Definition 2.17 (Lipschitz continuity)**.** *Function $f : \mathbb{R} \to \mathbb{R}$ is* Lipschitz *with* Lipschitz constant $L \geq 0$ *iff for every $x, y \in \mathbb{R}$, $\left|f(y) - f(x)\right| \leq L\left|y - x\right|$.*

Lipschitz is related to other forms of continuity as well as differentiability. In particular, every Lipschitz continuous function is indeed continuous. Moreover, it is *absolutely continuous* that makes it *differentiable almost everywhere*. In addition its derivative in absolute value is almost everywhere bounded by L. Hence the result of *fundamental theorem of calculus* may applies, ensuring that the function coincides with the Lebesgue integral of its derivative. This is specially useful when the function is characterised by a differential equation.

2.4.2 Convergence of sequences of functions

It is often useful to study functions not independently, but as a *sequence* or *series*. For a quantity that can be computed iteratively, the sequence is constructed by the value of quantity up to the n-th iteration. For instance, the cumulative reward in MRAs is gained step by step. This constitutes a sequence of functions, each corresponding to the cumulative rewards up to the n-th step. The overall cumulative reward is then the limit of this sequence. In most of the cases the limit is the quantity that we aim to compute or study. It is usually easier to establish properties for the functions in the sequence than its limit. Hence, we are interested to check if a property can be lifted from the sequence into its limit.

This section discusses the convergence of sequences of functions. There are different forms of convergences, among them we study *pointwise*, *monotone* and *uniform* convergences. We also discuss in these cases how limit processes like lim, inf, and sup can be interchanged. We also mention the properties that are preserved under a certain notion of convergence.

A sequence of functions. We consider a collection of functions that share the same domain and co-domain. For our purpose, we restrict to extended real-valued functions and do not consider more general forms of functions. However most of the results extend without difficulty to more general cases. A *sequence of function* is formed by a collection $f_1, f_2, \ldots$ of functions from an arbitrary set X to $\overline{\mathbb{R}}$, written as $\{f_n : X \to \overline{\mathbb{R}}\}_{n \in \mathbb{N}}$. It may converges to another function, which is called the *limit* of the sequence. Sometimes we need to be more descriptive and state also the form of convergence.

Pointwise convergence. One of the weakest way in which a sequence of functions converges to a particular function is called *pointwise convergence*. Without any further restriction, it states that for every point in the shared domain, the sequence is convergent.

Definition 2.18 (Pointwise convergence). *Sequence $\{f_n : X \to \overline{\mathbb{R}}\}_{n\in\mathbb{N}}$ is said to be* pointwise convergent *to $f : X \to \overline{\mathbb{R}}$, iff $\lim_{n\to\infty} f_n(x) = f(x)$ for every $x \in X$.*

We have already stated in Thm. 2.9 that pointwise convergence preserves Borel measurability. Apart from measurability, Lipschitz continuity is also preserved under pointwise convergence, provided that there is a global Lipschitz constant for all f_n's. As a result, f is indeed Lipschitz with the same constant. We will prove this result in our specific setting in Thm. 5.7 on page 95.

Monotone convergence. Monotone convergence entails the sequence of functions $\{f_n : X \to \overline{\mathbb{R}}\}_{n\in\mathbb{N}}$ being monotone, i. e. either *decreasing* or *increasing*; that is to say, $f_n(x) \geq f_{n+1}(x)$ (or respectively $f_n(x) \geq f_{n+1}(x)$) for all n and x. In addition, if the sequence is convergent to some function f, we shall say that the convergence is *monotone*. In other words, monotone convergence is stronger than pointwise convergence as it restricts the sequence to be additionally increasing or decreasing.

Assume the functions are defined on a set accompanied by a measure space; an important feature of monotone convergence is that it enables to interchange between limit and integral. It implies that the integral of the functions in the sequence with respect to the measure converges to the integral of the limit. In fact, the function are required to be Borel measurable. This result, known as *monotone convergence theorem*, is shown next for nonnegative real-valued functions.

Theorem 2.19 (Monotone convergence). *Let $(\mathbb{X}, \mathcal{X}, \mu)$ be a measurable space and $\{f_n : \mathbb{X} \to \overline{\mathbb{R}}_{\geq 0}\}_{n\in\mathbb{N}}$ be a sequence of Borel measurable functions that is monotonically convergent to f. It holds that*

$$\lim_{n\to\infty} \int_{\mathbb{X}} f_n \, \mathrm{d}\mu = \int_{\mathbb{X}} f \, \mathrm{d}\mu$$

Proof. See the proof of [AD99, Thm. 1.6.2]. □

Notice that the result of the monotone convergence theorem holds also for summation instead of integration. As an special case, this happens when $\mathbb{X}$ is countable and μ is a counting measure, a measure that counts the number of elements of a set.

Uniform convergence. Often enough in mathematical analysis, we want to check whether or not a property that holds for each function in a sequence can be carried over into its limit. For example, we may want to know if a sequence of continuous or differentiable functions converges to a continuous, or respectively differentiable function. To answer such questions, in many cases, we need stronger notions of convergence than pointwise convergence. It usually includes pointwise convergence together with more restrictions on the way the sequence converges. For example, a condition can be imposed on the speed of convergence. In this way, one may require that the sequence converges roughly with the same speed for all points in the domain. This leads to the concept of *uniform convergence*.

Definition 2.20 (Uniform convergence). *Given set X, then a sequence of functions $\{f_n : X \to \overline{\mathbb{R}}\}$ is* uniformly convergent *to $f : X \to \overline{\mathbb{R}}$ if and only if*

- *for every $\epsilon > 0$ there exist $N \in \mathbb{N}$ such that for all $n \geq N$ and all $x \in X \setminus (X_\infty \uplus X_{-\infty})$ we have $\left|f_n(x) - f(x)\right| < \epsilon$,*
- *for every $M \in \mathbb{R}$ there exist $N \in \mathbb{N}$ such that for all $n \geq N$ and all $x \in X_\infty$ we have $f_n(x) > M$,*
- *for every $M \in \mathbb{R}$ there exist $N \in \mathbb{N}$ such that for all $n \geq N$ and all $x \in X_{-\infty}$ we have $f_n(x) < M$,*

where $X_{\pm\infty} = \{x \in X : f(x) = \pm\infty\}$.

Uniform convergence makes sure that for sufficiently large N, we have all f_n with $n \geq N$ uniformly in ϵ-neighbourhood of f, or in the case of converging to infinity, uniformly larger than M (smaller than M in case the limit goes to $-\infty$). This brings in nice properties, for instance preservation of continuity. Moreover, one can interchange between limit and supremum/infimum operators taken over a uniformly convergent sequence.

Theorem 2.21. *Suppose X is a set and $\{f_n : X \to \overline{\mathbb{R}}\}$ is a sequence of functions converging uniformly to $f : X \to \overline{\mathbb{R}}$, then for* $\mathrm{opt} \in \{\inf, \sup\}$

$$\lim_{n\to\infty} \operatorname*{opt}_{x\in X} f_n(x) = \operatorname*{opt}_{x\in X} f(x)$$

Proof. We show the theorem for the case when $-\infty < M := \mathrm{opt}_{x\in X} f(x) < \infty$. Therefore, we can ignore sub-domains $X_{\pm\infty}$ and focus on $X \setminus (X_{-\infty} \uplus X_\infty)$. The infinity cases can be treated similarly. Fix $\epsilon > 0$; we have then all f_n's uniformly in ϵ-neighbourhood of f for n large enough. That is to say, there exist $N \in \mathbb{N}$ such that for all $n \geq N$ and $x \in X \setminus (X_{-\infty} \uplus X_\infty)$

$$\left|f_n(x) - f(x)\right| \leq \epsilon \tag{2.1}$$

At this point we distinguish between the supremum and infimum cases. From Eq. (2.1) for the supremum case we can obtain $f_n(x) \leq f(x) + \epsilon \leq \sup_{x\in X} f(x) +$

$\epsilon = M + \epsilon$. By taking the supremum again we have $\sup_{x\in X} f_n(x) \leq M + \epsilon$. The other side of inequality (2.1) gives $f(x) - \epsilon \leq f_n(x) \leq \sup_{x\in X} f_n(x)$ and thereby $M - \epsilon \leq f_n(x) \leq \sup_{x\in X} f_n(x)$. All together we can conclude

$$\left| \sup_{x\in X} f_n(x) - M \right| \leq \epsilon$$

With a similar argument we can obtain a similar inequality for the infimum case. Hence, for all $n \geq N$ and x it holds that

$$\left| \operatorname*{opt}_{x\in X} f_n(x) - M \right| \leq \epsilon$$

which completes the proof. □

As a special case, whenever set X is finite, we can conclude that the convergence is uniform. This is because the speed of convergence is uniformly greater than the speed of the convergence for the slowest point in the finite domain. Therefore, the result of Thm. 2.21 holds. This is summarised in the next theorem.

Theorem 2.22. *Suppose X is a finite set and $\{f_n : X \to \overline{\mathbb{R}}\}$ is a sequence of functions converging pointwise to $f : X \to \overline{\mathbb{R}}$, then for* $\text{mix} \in \{\min, \max\}$

$$\lim_{n\to\infty} \operatorname*{mix}_{x\in X} f_n(x) = \operatorname*{mix}_{x\in X} f(x)$$

Proof. It is enough to show that the convergence is uniform since then the claim follows from Thm. 2.21. Thanks to the finiteness of X, it is possible to satisfy the conditions of uniform convergence given in Def. 2.20. For every $\epsilon > 0$ we choose $N_x \in \mathbb{N}$ with $x \in X \setminus (X_\infty \uplus X_{-\infty})$ such that $n \geq N_x$ implies $\left|f_n(x) - f(x)\right| \leq \epsilon$. Since X is finite, we can pick $N = \max\{N_x : x \in X \setminus (X_\infty \uplus X_{-\infty})\}$ for the first condition of uniform convergence. We proceed in the same way for the other two conditions. For every $M \in \mathbb{R}$ we choose $N_x \in \mathbb{N}$ with $x \in X_\infty$ (respectively, $x \in X_{-\infty}$) such that $n \geq N_x$ implies $f_n(x) > M$ (respectively, $f_n(x) < M$). Finally, taking $N_{\pm\infty} = \max\{N_x : x \in X_{\pm\infty}\}$ in the second and the third condition completes the proof. □

The results of Thm. 2.21 and 2.22 will be used to justify the interchange between limits and optimum operators in Ch. 4 and 5.

We close this section by listing a number of references for further reading. The first four chapters of [AD99] gives an in-depth background in measure and probability theory. It covers the most of topics discussed in this chapter and much more. In addition, sequences and series of functions and in particular uniform convergence and its relation to continuity and differentiability is discussed in [Rud76, Ch. 7]. And lastly, an excellent and brief introduction to a measure theoretic approach to probability theory is given by [Neu10, Ch. 1].

Chapter 3

Markov Automata

In this chapter we define Markov automata (MA) [EHZ10b; EHZ10a; DH11; DH13] which is the main model considered in this thesis. Markov automata constitute a nondeterministic and stochastic model subsuming a wide range of important formalisms, including interactive Markov chains, probabilistic automata, discrete- and continuous-time Markov chains and labelled transition system. Markov automata are compositional, making them applicable for evaluation and analysis of modular systems. Their application domain spans a diverse area in industry, engineering and research. For instance, they provide the complete semantics for industrial formalisms including Generalised stochastic Petri nets [Eis+13], dynamic fault trees [BCS10] and Architecture Analysis and Design Language (AADL) [Boz+11].

In this chapter, we first introduce Markov automata and then study their compositionality. We then provide their semantics in terms of *histories* and *paths*. We develop σ-algebras and probability measures over the set of histories and paths induced by Markov automata. And finally we introduce *expectation splitting*, an effective technique for simplifying expectation computation in MA framework.

3.1 Markov Automata

Markov automata provide a compositional behavioural model combining some of the most important features that are necessary for modelling industrial and engineering systems. They integrate *Probabilistic Automata* (PA) with *Continuous-Time Markov Chains* (CTMC) to inherit properties from both models. They are thereby capable of modelling nondeterministic behaviour and discrete probabilistic branching from PA and stochastic timing from CTMC, making them one of the most general models that exist in formal methods.

Similar to other automaton like structures, a collection of states and transitions between them constitute an MA. A state characterises an internal status of the system modelled by the MA within a certain period of its execution. At the beginning, the MA is in the *initial state*, from which its execution starts. By *exe-*

cuting transitions, the MA can move between its states. There are *action*-labelled, known also as *probabilistic* transitions, and *rate*-labelled, known also as *Markovian* transitions. An action represents the event associated with the execution of the corresponding transition. It is the main tool enabling the MA to communicate with its external environment. Markovian transitions, on the other hand, are mainly used for modelling the internal dynamics of the MA. They govern for instance the *sojourn time* between state changes. With this brief introduction we provide the formal definition of the MA.

Definition 3.1 (Markov automaton). *A* Markov automaton (MA) *is a tuple* $\mathcal{M} = \big(V, v_0, Act, T_\mathrm{P}, T_\mathrm{M}\big)$*, where*

- V *is a finite set of states, of which* $v_0 \in V$ *is the* initial state*;*
- *Act is a countable set of* actions*, including internal invisible action* τ*;*
- $T_\mathrm{P} \subseteq V \times Act \times \mathrm{Distr}(V)$ *is the* probabilistic transition relation*;*
- $T_\mathrm{M} \subseteq V \times \mathbb{R}_{>0} \times V$ *is the* Markovian transition relation*.*

We consider the sets V, T_P and T_M to be finite in this thesis. By $\gamma = (v, \alpha, \mu) \in T_\mathrm{P}$ we say γ is an *outgoing* or *emanating* transition from v. We may also say that γ is *available* at v. In such a case γ can *fire* or *execute* action α and evolves to one of its successor state according to distribution μ. Similar terminology is used for Markovian transition $\gamma = (v, \lambda, v') \in T_\mathrm{M}$ that is an outgoing transition from v to v' with rate λ.

We classify states of an MA according to their outgoing transitions. We refer by $T_\mathrm{P}(v) := \{(v, \alpha, \mu) \in T_\mathrm{P} : \alpha \in Act, \mu \in \mathrm{Distr}(V)\}$ to the set of *probabilistic outgoing transitions* from v, and similarly by $T_\mathrm{M}(v) := \{(v, \lambda, v') \in T_\mathrm{M} : \lambda \in \mathbb{R}_{>0}, v' \in V\}$ to the set of *Markovian outgoing transitions* from v. With this we partition state space of an MA into the set of *probabilistic* (V_P), *Markovian* (V_M), *hybrid* (V_H) and *deadlock* (V_D) states, where

- $v \in V_\mathrm{P}$ iff $T_\mathrm{P}(v) \neq \emptyset$ and $T_\mathrm{M}(v) = \emptyset$,
- $v \in V_\mathrm{M}$ iff $T_\mathrm{P}(v) = \emptyset$ and $T_\mathrm{M}(v) \neq \emptyset$,
- $v \in V_\mathrm{H}$ iff $T_\mathrm{P}(v) \neq \emptyset$ and $T_\mathrm{M}(v) \neq \emptyset$,
- $v \in V_\mathrm{D}$ iff $T_\mathrm{P}(v) = \emptyset$ and $T_\mathrm{M}(v) = \emptyset$.

Intuitively speaking the set of probabilistic (Markovian) states are those that have only probabilistic (Markovian) outgoing transitions. Hybrid states have transitions of both types, and deadlock states have no outgoing transition.

3.1.1 Interpretation of transitions

A transition conveys information about its source, its destination and its timing dynamics. This information determines when the transition fires and what happens after its execution. Here we give the interpretation for both probabilistic and Markovian transitions.

Probabilistic transitions carry the possibly visible part of an MA behaviour, i. e. actions. However, not all of the actions are visible, in particular $\tau \in Act$ is considered invisible, describing an internal behaviour of the system modelled by the MA. We make a distinction between *external actions*, which are visible and meant to be used for synchronisation in parallel composition with other models, and *internal actions* that are invisible. Accordingly, we call a transition *internal* (*external*) iff it carries an internal (external) action. Note that the terms are only applicable to probabilistic transitions as Markovian ones are not action-labelled. There is a difference between internal and external transitions in regard to timing. Internal transitions are autonomous meaning that they fire instantaneously and thereby their execution cannot be delayed. Conversely external transitions can be delayed by synchronisation in case the model is in communication with another MA. In such a case the transition may not execute its external action until the other MA is also ready for synchronisation.

As explained above the timing of external transitions depends on the MA's context, i. e. the environment in which the MA is placed. The environment may consist of several other MAs communicating with each other and with the MA. Conversely the timing of internal transitions is independent of the context. Hence, we interpret internal transitions first, and we discuss about external transitions later in Sec. 3.1.2, where we define compositionality. Let $\gamma = (v, \alpha, \mu) \in V_{\mathrm{P}}(v)$ be an internal transition, i. e. action α is internal, and suppose γ is the only outgoing transition of v. Then, at the moment the model is in v, the transition fires. This also implies that the time of staying at v, known as the *sojourn time*, is zero. Immediately afterwards the model evolves to one of the successors of v. The choice of the successor state is probabilistic. More precisely, state $v' \in V$ is selected as the successor of v with probability $\mu(v')$.

Markovian transitions behave stochastically. In contrast to internal transitions, they are not instantaneous. Moreover their delay, opposed to external transitions does not depend on the MA's context, but rather their *exit rate*. The exit rate of a Markovian transition determines how it fires randomly. It completely identifies the distribution of the sojourn time of the transition source. Let v be a Markovian state, then the rate between v and an arbitrary state u is defined as $\mathrm{rate}(v, u) := \sum\{\lambda : (v, \lambda, u) \in T_{\mathrm{M}}(v)\}$. The exit rate of v is then the sum of its outgoing rates, denoted by $\mathrm{E}(v) := \sum_{v' \in V} \mathrm{rate}(v, v')$. The time to leave state v by firing one of its outgoing transitions is exponentially distributed with the parameter equal to its exit rate. Following from the probability density function of exponential distributions, the probability to leave v within interval $[a, b]$ is given by $\int_a^b \mathrm{E}(v)\, \mathrm{e}^{-\mathrm{E}(v) \cdot t}\, \mathrm{d}t = \mathrm{e}^{-\mathrm{E}(v) \cdot a} - \mathrm{e}^{-\mathrm{E}(v) \cdot b}$. After leaving v the choice of the successor state is governed by the *branching probability distribution* of v, which selects v' as the successor of v with probability $\frac{\text{rate}(v,v')}{\text{E}(v)}$. As the reader may have noticed, the interpretation of a Markovian transition in an MA coincides with that of a state with the same exit rate in a CTMC.

The above explanation implicitly provides the interpretation of a probabilistic state with only one outgoing transition and a Markovian state. There are still questions remaining unanswered. For instance, how to interpret a probabilistic state with more than one outgoing transition, or a hybrid state with both Markovian and probabilistic transitions. The former is explained by means of *strategies* in Sec. 3.2.1. The latter is discussed in the context of *compositionality*, which is presented in the next section.

3.1.2 Compositionality

Compositionality is one of the key features of MAs, enabling them to be used for modular modelling of complex systems. MAs are essentially designed to be compositional. It means that the MA of a complex system consisting of several *components* can be constructed by composing the MAs of the individual components. This feature is of crucial importance since modelling each component separately and then composing them is much more effective and natural than modelling the whole system at once. In order to obtain a faithful model of the system its components must be able to cooperate and communicate with each other. As mentioned before the communication is done by *synchronisation* over a number of actions. In this section we go more deeply into the details of the synchronisation and in general what is called *parallel composition*.

Parallel composition of MAs conservatively generalises that of PAs [Seg95] and Interactive Markov Chains (IMC) [Her02]. It is defined as an operator and applied to two MAs sharing some *synchronisation actions*. This constitutes a model that is the result of running both of the MAs in parallel while they communicate via the synchronisation actions. The state space of the composed model is the Cartesian product of the state spaces of its components. As expected its initial state is a pair consisting of the initial states of its component, and its action space is the union of the action spaces of its components. Construction of its transitions, however, requires more elaborations, which we discuss here. Parallel composition of two MAs is formally defined via a family of binary operators indexed by synchronisation sets [EHZ10a; EHZ10b]. We recap the definition here.

Definition 3.2 (Parallel composition). *Given two MAs $\mathcal{M}' = \left(V', v'_0, Act', T'_{\text{P}}, T'_{\text{M}}\right)$ and $\mathcal{M}'' = \left(V'', v''_0, Act'', T''_{\text{P}}, T''_{\text{M}}\right)$ with* synchronisation set $A \subseteq (Act' \cap Act'') \setminus \{\tau\}$, *then the composition of $\mathcal{M}'$ and $\mathcal{M}''$ with respect to parallel operator $||_A$ is defined as $\mathcal{M}' ||_A \mathcal{M}'' = \left(V, v_0, Act, T_{\text{P}}, T_{\text{M}}\right)$, where $V = V' \times V''$, $v_0 = (v'_0, v''_0)$, $Act = Act' \cup Act''$, and T_{P} and T_{M} are the smallest relations complying with the rules depicted in Tab. 3.1.*

In the rules, Δ *refers to the Dirac distribution. Moreover, we write* $\mathrm{sl}_{v',v''}$ *as* $\mathrm{rate}(v',v') + \mathrm{rate}(v'',v'')$ *for* $v' \in V'$ *and* $v'' \in V''$*. For* $\mu' \in \mathrm{Distr}(V')$ *and* $\mu'' \in \mathrm{Distr}(V'')$*,* $\mu' ||_A \mu'' \in \mathrm{Distr}(V' \times V'')$ *denotes the composed distribution with respect to* $||_A$ *with* $\mu' \,||_A\, \mu''(v',v'') := \mu'(v') \cdot \mu''(v'')$*.*

Some observations can be made by the rules of Tab. 3.1. The two components execute their transitions synchronously when the transitions are labelled with the same action that is in the synchronisation set. Otherwise they interleave their transitions. Similarly, the interleaving happens between Markovian transitions. Due to the *memoryless property* of exponential distributions, one component can fires its Markovian transition with the same distribution, irregardless of the fact that the other component may or may not have executed its Markovian transition. In other words, whatever happens the distribution of firing the Markovian transitions is still exponential with exactly the same rate as before.

Parallel composition of two components is carried out over a synchronisation set. After that, those actions in the synchronisation set that are not required for further synchronisation can be made hidden. This operation is called *hiding* and simply done by relabelling those actions to τ.

Definition 3.3. *Let* $\mathcal{M} = \big(V, v_0, Act, T_{\mathrm{P}}, T_{\mathrm{M}}\big)$ *be an MA, then* hiding $\mathcal{M}$ *with respect to* $A \subseteq Act \setminus \{\tau\}$ *gives MA* $\mathcal{M} \setminus A = \big(V, v_0, Act \setminus A, T'_{\mathrm{P}}, T_{\mathrm{M}}\big)$*, where* T'_{P} *is the smallest relation satisfying the rules depicted in Tab. 3.2.*

After parallel composition and hiding explained, we are ready to give the interpretation of hybrid states. We draw a distinction between two kinds: hybrid state with and without external actions. The discussion of both kinds not only clarifies how hybrid states behave, but also provides a helpful transition into the next topic, which describes *open* and *closed* MAs.

Firstly, consider a hybrid state without any external action, i. e. its outgoing transitions are either τ-labelled or Markovian. The interpretation of such a state is straightforward and can be indeed explained without resorting to the concept of compositionality. Since τ-labelled transitions are instantaneous on the one

$$\frac{\big(v',\alpha,\mu'\big) \in T'_{\mathrm{P}} \quad \alpha \notin A}{\big((v',v''),\alpha,\mu' \,||_A\, \Delta_{v''}\big) \in T_{\mathrm{P}}} \qquad \frac{\big(v'',\alpha,\mu''\big) \in T''_{\mathrm{P}} \quad \alpha \notin A}{\big((v',v''),\alpha,\Delta_{v'} \,||_A\, \mu''\big) \in T_{\mathrm{P}}}$$

$$\frac{\big(v',\alpha,\mu'\big) \in T'_{\mathrm{P}} \quad \big(v'',\alpha,\mu''\big) \in T''_{\mathrm{P}} \quad \alpha \in A}{\big((v',v''),\alpha,\mu' \,||_A\, \mu''\big) \in T_{\mathrm{P}}}$$

$$\frac{\big(v'_1,\lambda,v'_2\big) \in T'_{\mathrm{M}} \quad v'_1 \neq v'_2}{\big((v'_1,v''),\lambda,(v'_2,v'')\big) \in T_{\mathrm{M}}} \qquad \frac{\big(v''_1,\lambda,v''_2\big) \in T''_{\mathrm{M}} \quad v''_1 \neq v''_2}{\big((v',v''_1),\lambda,(v',v''_2)\big) \in T_{\mathrm{M}}} \qquad \frac{\mathrm{sl}_{v',v''} > 0}{\big((v',v''),\mathrm{sl}_{v',v''},(v',v'')\big) \in T_{\mathrm{M}}}$$

Table 3.1: Structural operational semantics for the parallel composition defined in Def. 3.2.

$$\frac{\big(v,\alpha,\mu\big) \in T_{\mathrm{P}} \quad \alpha \notin A}{\big(v,\alpha,\mu\big) \in T'_{\mathrm{P}}} \qquad\qquad \frac{\big(v,\alpha,\mu\big) \in T_{\mathrm{P}} \quad \alpha \in A}{\big(v,\tau,\mu\big) \in T'_{\mathrm{P}}}$$

Table 3.2: Structural operational semantics of hiding defined in Def. 3.3.

hand and the probability of immediate execution of Markovian transitions is zero on the other hand, there is no chance for Markovian transitions to be fired before any τ-labelled transition. Consequently, in a hybrid state without any external actions, τ-labelled transitions are almost surly executed earlier than Markovian transitions. The concept is known as *maximal progress* assumption, requiring that in general, probabilistic transitions labelled with internal actions and Markovian transitions can not be executed at the same time. The former takes precedence over the latter, wherever both enabled, i. e. in hybrid states.

Now we turn our attention to the other kind: hybrid states with external actions. As it can be inferred from the definition of parallel composition, the execution time of an external action entirely depends on the context, i. e. what model is placed in parallel with the MA. This kind of MAs that have transitions labelled with external actions, and thereby can still interact with the other MAs via parallel composition are known as *open* MAs. Therefore the interpretation of an open MA, and in particular those hybrid states is not understood without the full knowledge of the environment in which the MA is situated.

The closest attempts to provide an interpretation of open MAs are perhaps [Brá+12; HKK13]. Their approach works for IMCs instead of MAs. They embed an open IMC into a timed game played with an unknown environment that controls the execution of the external actions. The evaluation of the game at the end provides a compositional verification of the IMC by establishing bounds on reachability probabilities. The bounds are guaranteed to hold under any circumstances regardless of the environment or the context in which the IMC is located. The approach has not been yet extended to be applicable for MAs. We are unaware of any other technique trying to put an interpretation on open MAs.

We conclude the discussion of this section by pointing out that this thesis does not provide the interpretation and analysis of open MAs. We instead turn our attention to *closed* MAs, which are the MAs that cannot be composed with other MAs, i. e. they cannot interact with their environment. Therefore their interpretation is understood without any extra knowledge of the context in which they are. We describe closed MAs and explain their properties in the next section.

3.1.3 Closed MAs

MAs can be locked up against interaction with their environment, meaning that they are not anymore considered subject to further composition. Such an MA is called *closed*. In a closed MA there is no external action enabled in the whole model whatsoever. Therefore, none of the actions can be delayed by synchronisation. Furthermore, each outgoing transition from any state is labelled with an internal action. Consequently, we can put a definite interpretation on each outgoing transition in the MA. They are either internal or Markovian. We can further remark on two points regarding closed MAs.

Remark 3.4. *It is clear that nothing prevents internal transitions from happening instantaneously. Since all enabled transitions of a closed MA are internal, we can impose maximal progress assumption to safely eliminate Markovian outgoing transitions of all hybrid states. As a result all hybrid states can be turned into probabilistic states without affecting their interpretation.*

Remark 3.5. *As the actions available in a closed MA are not subject to further interaction, they do not carry any meaning. They are just labels that can be safely removed. We then, instead of meaningless actions, label probabilistic transitions with $\natural$, signifying that they are fired instantaneously. Furthermore, we aggregate all outgoing transitions of each Markovian state into a single transition labelled with the exit rate of the state. The transition includes, in addition, the branching probability distribution of the state. This way of representation unifies the notation of probabilistic and Markovian transitions without altering their meaning.*

Following from the remarks we adapt Def. 3.1 for closed MAs.

Definition 3.6 (Closed Markov automaton)**.** *A closed MA is a triple $\mathcal{M} = (V, v_0, T)$, where V, v_0 are as described in Def. 3.1, and $T \subseteq V \times \mathbb{R}^{\natural}_{>0} \times \mathrm{Distr}(V)$ is the transition relation with $\mathbb{R}^{\natural}_{>0} = \mathbb{R}_{>0} \uplus \{\natural\}$.*

Furthermore, let, for every $v \in V$, $T_{\mathrm{P}}(v)$ be the set of probabilistic outgoing transitions *of v, i. e., $T_{\mathrm{P}}(v) := \{(v, \natural, \mu) \in T\}$, and similarly, $T_{\mathrm{M}}(v)$ be the set of* Markovian outgoing transitions *of v, i. e., $T_{\mathrm{M}}(v) := \{(v, \lambda, \mu) \in T \mid \lambda \in \mathbb{R}_{>0}\}$. Then, T* must *satisfies:*

- *for every $v \in V$, at least one of $T_{\mathrm{P}}(v)$ or $T_{\mathrm{M}}(v)$ is empty (Rem. 3.4);*
- *for every $v \in V$, $|T_{\mathrm{M}}(v)| \leq 1$ (Rem. 3.5).*

Once more we want to stress that Def. 3.6 complies with Def. 3.1 in terms of the interpretation developed in Sec. 3.1.1 and 3.1.2. It indicates that the special description of closed MAs depicted in Def. 3.6 does not alter their semantics compared with what is generally represented as MAs in Def. 3.1. There are, however, syntactic differences. Markovian and probabilistic transition relations are separately defined in Def. 3.1, but now in Def. 3.6 they are merged into a single set denoted by T. Therein, label $\natural$ of a probabilistic transition distinguishes it from a Markovian one, which carries instead a positive exit rate. Moreover, we make sure that there is no hybrid state in the model, as the maximal progress assumption is by default imposed. For technical reasons, we need to eliminate deadlock states as well. For this, we decorate each deadlock state with a Markovian self loop with an arbitrary rate. As a consequence, the model is stuck in deadlock states while time progresses.

We recapitulate the interpretation of Markovian and probabilistic transitions adapted to the new notation. Transition $(v, \lambda, \mu) \in T$ is probabilistic, if $\lambda = \natural$, otherwise ($\lambda \in \mathbb{R}_{>0}$) it is Markovian. If probabilistic, it will be executed instantaneously. Otherwise it is fired after an exponentially distributed random delay with parameter λ. The successor of v for both Markovian and probabilistic transitions is selected according to distribution μ. Consequently, for Markovian case, the probability that the transition is fired within $t \geq 0$ time units and the model moves to state $v' \in V$ is given by $\mu(v') \cdot (1 - \mathrm{e}^{-\lambda \cdot t})$.

As stated previously, we use $T_{\mathrm{M}}(v)$ and $T_{\mathrm{P}}(v)$ to refer to the set of Markovian and probabilistic transitions available at state v. Then, $T(v) = T_{\mathrm{M}}(v) \uplus T_{\mathrm{P}}(v)$ is the set of all transitions available at state v. We denote the set of Markovian and probabilistic transitions by T_{M} and T_{P}, respectively. The same state partitioning, as discussed before, prevails in this case with, indeed, the fact that the sets of hybrid and deadlock states are both empty. Given transition $\gamma = (v, \lambda, \mu) \in T$, we use functions $\mathsf{v}_\gamma := v$, $\mathsf{r}_\gamma := \lambda$ and $\mathsf{d}_\gamma := \mu$ to project γ into its source, its rate and its distribution, respectively.

Closedness of an MA enables it to autonomously move between its states via its transitions without interacting with the environment in which it is placed. Considering that its complete behaviour can be understood independent of any external objects, it is of interest as a subject for different kinds of analyses. As of now we assume all MAs that are subject to analysis are closed and, unless stated differently, the term MA refers to closed MA. We provide the complete semantics of [closed] MAs in the next section.

3.1.4 Paths and histories

By execution of an MA, starting from its initial state, it traverses its state space through firing its transitions. The execution of each transition follows its specific interpretation, as discussed previously. The successive execution of the transitions continues forever, exhibiting a sample behaviour of the MA. In transition system terminology this is usually known as *execution trace*, *trajectory* or *path*. The set of all paths that can be observed from an MA captures its complete dynamics. In this section we elaborate on different kinds of *traces*, which are *paths* and their finite forms *histories* and *path fragments*. Before going into their definition, their main building block, called *step*, is introduced here.

Definition 3.7 (Step). *A* step $\varsigma \in (T_{\mathrm{M}} \times \mathbb{R}_{\geq 0}) \uplus (T_{\mathrm{P}} \times \{0\})$ *is a pair of a transition and its execution time. The set of all steps is denoted by* $\mathbb{S}$.

A step taken in a closed MA is essentially interpreted according to its type of transition, as discussed in Sec. 3.1.3. Step $(\gamma, t) \in \mathbb{S}$ indicates that the MA resides in state v_γ for t time units before executing γ. In other words, the sojourn time in v_γ is t time units. In case the transition is probabilistic, it must be taken immediately, i. e. $t = 0$. For a Markovian transition, the sojourn time can however be any nonnegative value. Steps act as building blocks of other trace-like structures such as histories. Here, we explain how the concatenation of multiple steps produces a history.

Definition 3.8 (History). *A* history $\eta \in \mathbb{S}^* \times V$ *is a finite sequence of steps ending in a state.*

A history of an MA contains a finite sequence of steps followed by the *last* state, where the history meets the present. The MA has visited all of the states in the history up to the last state. In this case, we say that the history is executed by the MA. The interpretation of the whole history is obtained by assembling the interpretation of all the steps and that of its last state. That is, the MA executes the transitions of its steps one by one at the specified time points until it reaches the last state.

We introduce here some useful notations regarding histories. We refer to $|\eta|$ as the length of history η, which is the number of its transitions. The last state of η is denoted by $\eta\downarrow$. We usually write $\gamma \xrightarrow{t}$ instead of step (γ, t) and thus represent a history as $\eta = \gamma_0 \xrightarrow{t_0} \gamma_1 \xrightarrow{t_1} \cdots \gamma_{n-1} \xrightarrow{t_{n-1}} v$. Taking history η into consideration, we can refer to elements of η by their index. Given index $i \in \{0, \ldots, n-1\}$, γ_i, which is the $(i+1)$-th transition of η, is denoted by Γ_i^η. We can refer to the source, the rate and the distribution of Γ_i^η by v_i^η, r_i^η and d_i^η, respectively. Moreover, we use $\eta\langle i\rangle$ to denote t_i, the $(i+1)$-th sojourn time in v_i^η.

A path extends a history by continuing its sequence of steps forever.

Definition 3.9 (Path). *A* path $\pi \in \mathbb{S}^\omega$ *is an infinite sequence of steps.*

Similar to the histories, a path is represented as $\eta = \gamma_0 \xrightarrow{t_0} \gamma_1 \xrightarrow{t_1} \cdots$. Its length is indeed infinity and its last state is undefined. Apart from that, other notations that are used for histories can also be employed for paths in a straightforward way.

For technical reasons, we need to define path fragments separately from histories. Intuitively, a fragment is a history whose last state is left out. Thus the last state of a fragment is unspecified.

Definition 3.10 (Fragment). *A* path fragment*, or simply* fragment *is a finite sequence of steps, namely an element of set* $\mathbb{S}^*$.

All the concepts and notations used for histories, except for the last state are carried over to fragments. Furthermore, we define the *null fragment* to be the fragment with length zero, denoted by $\varepsilon\epsilon$.

We use operator $\circ$ to concatenate a fragment to a trace. As stated before a trace can be a fragment, a history or a path. Given fragment φ and trace x, then the trace generated by the concatenation of φ and x, is denoted by $\varphi \circ x$. The concatenation is naturally done by attaching x to φ such that the first transition of x is directly executed after the last transition of φ. It obviously holds that $\varepsilon \circ x = x$. It is then clear that $\varphi \circ x$ is of the same type as x, for instance $\varphi \circ \pi \in \mathbb{P}$, if $\pi \in \mathbb{P}$.

It is often useful to extract a sub-fragment, sub-history or sub-path lying in a trace up to or starting from a specific step. For trace x that has at least $n \geq 0$ step(s), $\lceil x \rceil^n$ is referred to as the prefix of x up to the n-th step. A sub-fragment or a sub-history might be considered as the prefix. If the prefix is intended to be a sub-fragment, it will be obtained by taking the first n step(s) of x. The sub-history, as expected, is the sub-fragment followed by v_n^x, as the last state. In the sequel, it will become clear by the context whether a sub-fragment or a sub-history is meant. The initial state of x can be obtained by sub-history $\lceil x \rceil^0$, and also sub-fragment $\lceil x \rceil^0$ is nothing but ε. The suffix of x after (but not including) the n-th step, $\lfloor x \rfloor_n$, is extracted by removing the first n step(s) of x, if existing. Here, again the trace and its suffix are of the same type. It holds that $\lfloor x \rfloor_0 = x$.

Fragments are technically important since every path $\pi \in \mathbb{P}$ can be split into a sub-fragment $\lceil \pi \rceil^n$ and a sub-path $\lfloor \pi \rfloor_n$ for all $n \in \mathbb{N}$ such that $\pi = \lceil \pi \rceil^n \circ \lfloor \pi \rfloor_n$. The probability measure and, more importantly, other objectives discussed in this thesis can be split accordingly. More details are found in Sec. 3.2 and 3.3.

Given MA $\mathcal{M}$, we use $\mathbb{F}^{\mathcal{M}}$, $\mathbb{H}^{\mathcal{M}}$ and $\mathbb{P}^{\mathcal{M}}$ to denote the set of all fragments, histories and paths of $\mathcal{M}$, respectively. Furthermore, for $n \in \mathbb{N}$, $\mathbb{F}_n^{\mathcal{M}}$ and $\mathbb{H}_n^{\mathcal{M}}$ are referred to as the set of all fragments and histories of length n, respectively. We omit superscript $\mathcal{M}$ whenever it is clear from the context.

The next section introduces measurable spaces for MAs, which are crucially based upon the concept histories and paths.

3.2 Measurability for MAs

A pattern of an MA's behaviour is specified by a collection of paths. Intuitively speaking, such a pattern, which is also called an *event*, is measurable if its probability can be measured. We study measurability to figure out for what kind of events in MAs the probability can be assessed. Understanding measurability is essential for the analysis of MAs. Notably, a particular analysis is measurable, provided that it involves a collection of events that are all measurable. In mathematical analysis *measurable spaces* are used to identify such collections. This section is devoted to defining measurable spaces for the set of fragments, histories and paths of an MA.

A collection of measurable events that is closed under infinitely many set operations is referred to as a σ*-algebra* (cf. Def. 2.1). The purpose of this section is to provide the formal definition of σ-algebras over the set of fragments, histories and paths. The construction of σ-algebras can be done in a modular way. Having σ-algebras defined on two sets X_1 and X_2, it is possible to extend them into a σ-algebra on their product space $X_1 \times X_2$, based on the concept of *product σ-algebra* (cf. Def. 2.11). In our setting for instance, a fragment is the product of a finite number of steps; hence the σ-algebra over the set of fragments can be built up from the σ-algebra over the set of steps.

A step is defined as a pair of a transition and its execution time. As the set of all steps is the disjoint union of steps on Markovian and probabilistic transitions, we construct the σ-algebra over each set separately. For the steps on Markovian transitions we employ the construction of the product σ-algebra. It is the smallest σ-algebra containing all *measurable rectangles* as the basic events. Recall that a measurable rectangle (Def. 2.10) is a tuple of measurable sets, each coming from the respective σ-algebra. For steps taken by Markovian transitions, it is of the form $\Gamma \times B$ where $\Gamma \in 2^{T_\mathrm{M}}$ and $B \in \mathcal{B}(\mathbb{R}_{\geq 0})$ with 2^{T_M} being the power set of T_M, and $\mathcal{B}(\mathbb{R}_{\geq 0})$ the Borel σ-algebra over nonnegative reals. The smallest σ-algebra containing all of those measurable rectangles is denoted by $2^{T_\mathrm{M}} \otimes \mathcal{B}(\mathbb{R}_{\geq 0})$. For probabilistic transitions the execution time is always zero, therefore the power set of $T_\mathrm{P} \times \{0\}$, denoted by $2^{T_\mathrm{P} \times \{0\}}$ is the appropriate σ-algebra. As a result $\mathcal{S}$, the σ-algebra over the set of steps, is defined as $\bigcup_{S_p \in 2^{T_\mathrm{P} \times \{0\}}} \left(2^{T_\mathrm{M}} \otimes \mathcal{B}(\mathbb{R}_{\geq 0}) \cup S_p\right)$.

The basis of the σ-algebra over the set of fragments, histories and paths is formed by $\mathcal{S}$, again using product σ-algebra. Accordingly, the σ-algebra over the set of fragments of length $n \geq 1$ is given by $\mathcal{F}_n := \bigotimes_{i=1}^{n} \mathcal{S}$. Moreover, $\mathcal{H}_n := \mathcal{F}_n \otimes 2^V$ is the σ-algebra over the set of histories of length n. For the set of paths, the σ-algebra is obtained by applying standard cylinder set construction (cf. [Neu10, Sec. 2.5.4]). Briefly speaking, let B_n be a subset of $\mathbb{H}_n$, then the cylinder of base B_n is described as $\mathrm{Cyl}(B_n) := \{\pi \in \mathbb{P} \mid \lceil \pi \rceil^n \in B_n\}$. A cylinder is *measurable* iff its base is measurable; for instance, $\mathrm{Cyl}(B_n)$ is measurable iff $B_n \in \mathcal{H}_n$. Then, $\mathcal{P}$, the σ-algebra over the set of paths, is the smallest σ-algebra generated by the class of all measurable cylinders i. e. $\bigcup_{i=0}^{\infty}\{\mathrm{Cyl}(B_n) \mid B_n \in \mathcal{H}_n\}$ (cf. [Neu10, Lem. 2.10]).

The above mentioned σ-algebras provide measurable spaces over the set of steps $(\mathbb{S}, \mathcal{S})$, also over the set of traces[1] $(\mathbb{F}_n, \mathcal{F}_n)$, $(\mathbb{H}_n, \mathcal{H}_n)$ and $(\mathbb{P}, \mathcal{P})$ for $n \in \mathbb{N}$. The spaces are used to reason the measurability of events and functions related to the dynamics of MAs. Moreover, by using a *measurable strategy*, which is discussed in the next section, we establish a probability measure on each of the measurable spaces.

3.2.1 Strategies

MAs may exhibit nondeterminism occurring at a state with two or more outgoing transitions. In such a state it is not clear a priori which transition is taken during the execution of the model. There must be an external object, called *strategy*[2] to resolve the nondeterminism. In the absence of nondeterminism, which has

[1] Note that $\mathbb{F}_0 = \{\varepsilon\}$ and $\mathcal{F}_0 = \{\emptyset, \mathbb{F}_0\}$.

[2] It is also known as policy, scheduler or adversary.

been resolved by strategies, MAs exhibit pure stochastic behaviour. As a result, a probability measure can be defined on the measurable space related to an MA. In other words, a complete semantics as a stochastic process can be delivered to an MA that is under the governance of a strategy.

Strategies use the details of histories such as the transitions, their order and their execution time to resolve nondeterminism. They are classified according to the amount of details they employ. The most general class exploits the complete information from histories to decide between transitions available at their last states. This class is further pruned by considering only the strategies that are measurable. It is known as the class of *generic measurable strategies* [WJ06; Joh08; Neu10], which is defined next.

Definition 3.11 (Generic measurable strategy)**.** *A* generic strategy *of MA $\mathcal{M}$ is a function, $\sigma : \mathbb{H} \to \mathrm{Distr}(T)$, such that for every $\eta \in \mathbb{H}$ it holds that* $\mathrm{supp}(\sigma(\eta)) \subseteq T(\eta{\downarrow})$*. Strategy σ is* generic measurable *iff $\{\eta \in \mathbb{H}_n \mid \sigma(\eta)(\gamma) \in B\} \in \mathcal{H}_n$ for every $\gamma \in T$, $B \in \mathcal{B}([0,1])$ and $n \in \mathbb{N}$. We use Σ to denote the set of all generic measurable strategies.*

The support restriction in Def. 3.11 indicates that for every history η, the strategy can only select the outgoing transitions available at the last state of η. Precisely, if $\sigma(\eta)(\gamma) > 0$, then it must hold that $\gamma \in T(\eta{\downarrow})$. Measurability of a strategy intuitively means that it never resolves nondeterminism for histories that in any way induce non-measurable sets. That is to say, the preimage of the strategy under every subset of transitions and every Borel set in $[0,1]$ can be partitioned into sets, each containing histories of the same length that belongs to the respective σ-algebra. As the probability measure is defined ultimately over the set of paths, we are interested in the *extension* of the preimage to paths. The extension of set of histories $H \subseteq \mathbb{H}$ is done by extending every history in H to a path in every possible way, that is $\mathrm{ext}(H) = \{\pi \in \mathbb{P} \mid \lceil \pi \rceil^{|\eta|} = \eta, \eta \in H\}$. A direct consequence of our definition ensures that the extension of the preimage is in $\mathcal{P}$.

Lemma 3.12. *For every measurable strategy σ, $\gamma \in T$ and $B \in \mathcal{B}([0,1])$ it holds that* $\mathrm{ext}\big(\{\eta \mid \sigma(\eta)(\gamma) \in B\}\big) \in \mathcal{P}$.

Proof. The lemma directly follows from the fact that the extension is the countable union of cylinders whose bases are measurable by definition, namely

$$\mathrm{ext}\big(\{\eta \mid \sigma(\eta)(\gamma) \in B\}\big) = \bigcup_{n=0}^{\infty} \mathrm{Cyl}\big(\{\eta \in \mathbb{H}_n \mid \sigma(\eta)(\gamma) \in B\}\big)$$

□

Our definition of measurability follows that of [Fu14a]. It is in contrast to [WJ06; Neu10], which defines the measurability with respect to the measurable space induced by the set of histories, i. e. $(\mathbb{H}, \mathcal{F}_{\mathbb{H}})$. However, it is not possible to define any meaningful measure on that measurable space[3]. Therefore, we avoid the definition of measurability in that way.

Building upon strategies

There are situations where we need to develop a strategy from another strategy. This is useful, in particular, when we want to force a strategy to decide in a specific way for a set of histories. The need for this strategy emerges when we want to change the decision of the old strategy only for a specific history or want to mimic its decision when we know a fragment has happened. Those situations frequently arise when analysing a model. We exploit them in proofs, specially in Ch.4 and 5. Let us first define a strategy that requires a specific transition to be always selected whenever starting from its source state.

Definition 3.13. *Given a measurable strategy $\sigma \in \Sigma$, then $\sigma|_\gamma$ with $\gamma \in T$ is defined as:*

$$\sigma|_\gamma(\eta) := \begin{cases} \Delta_\gamma & \eta = \mathsf{v}_\gamma \\ \sigma(\eta) & \eta \neq \mathsf{v}_\gamma \end{cases}$$

Intuitively speaking, $\sigma|_\gamma$ mimics the decisions of σ for all histories except for v_γ, in which transition γ is selected with probability one. It can be shown that $\sigma|_\gamma$ is measurable.

Lemma 3.14. *For all $\sigma \in \Sigma$ and $\gamma \in T$, $\sigma|_\gamma$ is measurable.*

Proof. For $\gamma' \in T$, $B \in \mathcal{B}([0,1])$ and $n \in \mathbb{N}$ we define sets $X_n^{B,\gamma'} := \{\eta \in \mathbb{H}_n \mid \sigma(\eta)(\gamma') \in B\}$ and $Y_n^{B,\gamma'} := \{\eta \in \mathbb{H}_n \mid \sigma|_\gamma(\eta)(\gamma') \in B\}$. In case $n = 0$, $\gamma = \gamma'$ and $1 \in B$, $Y_n^{B,\gamma'} = X_n^{B,\gamma'} \cup \{\mathsf{v}_\gamma\}$, otherwise $Y_n^{B,\gamma'} = X_n^{B,\gamma'}$. The claim then follows from $X_n^{B,\gamma'} \in \mathcal{H}_n$. □

Sometimes it is useful to construct a new strategy that presumes a fragment has just been visited and then complies with another strategy for forthcoming decisions.

Definition 3.15. *Given a measurable strategy $\sigma \in \Sigma$ and a fragment $\varphi \in \mathbb{F}$, then $\sigma[\varphi]$ resolves nondeterminism for history η as σ does it for $\varphi \circ \eta$, i. e. $\sigma[\varphi](\eta) := \sigma(\varphi \circ \eta)$.*

It is supposed by strategy $\sigma[\varphi]$ that every history is preceded by fragment φ. By taking it into account, for each history η the strategy mimics the decision of σ for $\varphi \circ \eta$. It can be shown that resolving nondeterminism in this way leads to a measurable strategy.

[3] The σ-algebra over the set of histories is the σ-algebra generated by collection $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$. The elements of the collection are pairwise disjoint. Two events $H_1 \in \mathcal{H}_{n_1}$ and $H_2 \in \mathcal{H}_{n_2}$ with $n_1 \neq n_2$ are considered as disjoint since their lengths are different. Nevertheless the events might share some common stems making their extension overlapping.

Lemma 3.16. *For every $\sigma \in \Sigma$ and $\varphi \in \mathbb{F}$, $\sigma[\varphi]$ is measurable.*

Proof. Let $X_n^{B,\gamma'} := \{\eta \in \mathbb{H}_n \mid \sigma(\eta)(\gamma') \in B\}$ and $Y_n^{B,\gamma'} := \{\eta \in \mathbb{H}_n \mid \sigma[\varphi](\eta)(\gamma') \in B\}$ for some $\gamma' \in T$, $B \in \mathcal{B}([0,1])$ and $n \in \mathbb{N}$. Then, from the measurability of σ, it holds that $X_{n+|\varphi|}^{B,\gamma'} \cap (\{\varphi\} \times \mathbb{H}_n) \in \mathcal{H}_{|\varphi|+n}$. On the other hand, the set equals $\{\varphi\} \times Y_n^{B,\gamma'}$. Therefore, it must hold that $Y_n^{B,\gamma'} \in \mathcal{H}_n$, otherwise it contradicts the measurability of σ. □

As a special case we are interested in situations where $\varphi = \gamma \xrightarrow{t}$ with $\gamma \in T$ and $t \in \mathbb{R}_{\geq 0}$. In order to resolve nondeterminism for history η in this case, $\sigma[\gamma \xrightarrow{t}]$ assumes that γ is fired at time t before η is visited. We use this strategy in Ch. 5 for proving the fixed point characterisation of the optimal ERR (Thm. 5.2).

Early vs. late strategies

In this thesis we consider *early* strategies, which are the only possible strategies that exist in MA's world. The difference between early and *late* strategies prevails in the situations where there are more than one Markovian transition available in a state. In this case, early [NSK09] means that the decision of which transition to take has to be made when entering a state and it may not be changed while residing in the state. On the contrary, a late decision may be changed at any time while staying in the state. In an MA, Markovian states have only one outgoing transition, as a result making a late or early decision induces the same result. Therefore late strategies are not relevant for MAs. In contrast to MAs, early and late strategies produce difference results in the analysis of CTMDPs, since having a state with two Markovian transitions is allowed there. For more details see [NSK09; Neu10]

3.2.2 Probability measure

The purpose of a probability measure is to assess how likely it is for an MA to observe an event. It basically determines the probability of executing a measurable set of paths starting from a given state while a measurable strategy is set. Similar to the construction of σ-algebra, we first construct the probability measure on the set of steps, and then extend it to more complex traces, like paths. Moreover we show that our construction always lead to a unique probability measure. The probability measure on the set of steps is defined as follows.

Definition 3.17 (Probability meausre on steps)**.** *Let $(\mathbb{S}, \mathcal{S})$ be the measurable space over the set of steps, as defined on page 39. Assume further that $\sigma \in \Sigma$ is a strategy of MA $\mathcal{M}$ and $\varphi \in \mathbb{F}$. Then, $\mu_\sigma(\varphi, \cdot) : \mathcal{S} \mapsto [0,1]$ is defined as*

$$\mu_\sigma(\varphi, S) := \sum_{\{\gamma \mid (\gamma,t) \in S\}} \begin{cases} \mathsf{d}_{|\varphi|-1}^{\varphi}(\mathsf{v}_\gamma) \cdot \sigma[\varphi](\mathsf{v}_\gamma)(\gamma) \cdot P_{\mathrm{ex}}^{S}(\gamma) & \varphi \neq \varepsilon \\ \sigma(\mathsf{v}_\gamma)(\gamma) \cdot P_{\mathrm{ex}}^{S}(\gamma) & \varphi = \varepsilon \end{cases}$$

with

$$P_{\mathrm{ex}}^{S}(\gamma) := \begin{cases} \displaystyle\int_{\mathbb{R}_{\geq 0}} \mathbb{1}_S(\gamma, t) \cdot \mathsf{r}_\gamma \cdot \mathrm{e}^{-\mathsf{r}_\gamma t}\, \mathrm{d}t & \gamma \in T_{\mathrm{M}} \\ \mathbb{1}_S(\gamma, 0) & \gamma \in T_{\mathrm{P}} \end{cases}$$

Intuitively speaking, $\mu_\sigma(\varphi, S)$ measures the probability of executing steps in S given that the steps are preceded by φ and strategy σ is set. For each transition γ that appears in S, it measures the probability of its execution at some time point in $\{t \mid (\gamma, t) \in S\}$. It makes a difference whether φ is the null fragment or not. In the latter case, the probability is computed in three parts:

1. $\mathsf{d}_{|\varphi|-1}^{\varphi}(\mathsf{v}_\gamma)$: the probability to land in the source state of γ after executing the last transition of φ,

2. $\sigma[\varphi](\mathsf{v}_\gamma)(\gamma)$: the probability that σ selects γ at v_γ when fragment φ has just been visited,

3. $P_{\mathrm{ex}}^{S}(\gamma)$: the probability to execute γ according to the sojourn times given in S.

It is obviously a fact that, when $\varphi = \varepsilon$, the term $\mathsf{d}_{|\varphi|-1}^{\varphi}(\mathsf{v}_\gamma)$ is not applicable and also $\sigma[\varphi](\mathsf{v}_\gamma)(\gamma) = \sigma(\mathsf{v}_\gamma)(\gamma)$. For the third term we distinguish between Markovian and probabilistic transitions. In case γ is Markovian, the probability of its execution is governed by the exponential distribution as discussed in Sec. 3.1.3. Otherwise, γ, as a probabilistic transition, is fired immediately.

All the three terms that contribute in the measure are probability distributions. It therefore holds that $\mu_\sigma(\varphi, \cdot)$ is a probability measure. To put it formally, we consider the measure as the extension of sub-measures on the product of transitions and sojourn times. The result is stated in the next lemma.

Lemma 3.18. *For any $\varphi \in \mathbb{F}$ and $\sigma \in \Sigma$, $\mu_\sigma(\varphi, \cdot) : \mathcal{S} \mapsto [0, 1]$ is the unique probability measure on $(\mathbb{S}, \mathcal{S})$.*

Proof. By employing product measure theorem (Thm. 2.12) we show that $\mu_\sigma(\varphi, \cdot)$ as described in Def. 3.17 is indeed a probability measure and is also unique. For that we define measure $\mu_1^{\varphi,\sigma} : 2^T \to [0, 1]$ such that

$$\mu_1^{\varphi,\sigma}(\{\gamma\}) := \begin{cases} \mathsf{d}_{|\varphi|-1}^{\varphi}(\mathsf{v}_\gamma) \cdot \sigma[\varphi](\mathsf{v}_\gamma)(\gamma) & \varphi \neq \varepsilon \\ \sigma(\mathsf{v}_\gamma)(\gamma) & \varphi = \varepsilon \end{cases}$$

for $\gamma \in T$. It is intuitively the probability to execute γ right after fragment φ. Furthermore, we define $\mu_2(\gamma, \cdot) : \mathcal{B}(\mathbb{R}_{\geq 0}) \to [0, 1]$ with

$$\mu_2(\gamma, B) := \begin{cases} \int_B \mathsf{r}_\gamma \cdot \mathrm{e}^{-\mathsf{r}_\gamma t}\, \mathrm{d}t & \gamma \in T_{\mathrm{M}} \\ \mathbb{1}_B(0) & \gamma \in T_{\mathrm{P}} \end{cases}$$

This intuitively measures the probability of executing γ at some time point within B. It is not hard to see that $\mu_1^{\varphi,\sigma}$ and $\mu_2(\gamma,\cdot)$ are measures; they are both non-negative and countably additive, and give the zero measure to empty set. To see countable additivity (cf. Def. 2.3) of $\mu_2(\gamma,\cdot)$ for $\gamma \in T_{\mathrm{M}}$, note that for any pairwise disjoint collection $\{B_n \in \mathcal{B}(\mathbb{R}_{\geq 0})\}_{n\in\mathbb{N}}$ and $N \in \mathbb{N}$:

$$\mu_2(\gamma, \biguplus_{n=0}^{N} B_n) = \int_{\biguplus_{n=0}^{N} B_n} \mathrm{r}_\gamma \cdot \mathrm{e}^{-\mathrm{r}_\gamma t}\,\mathrm{d}t = \sum_{n=0}^{N} \int_{B_n} \mathrm{r}_\gamma \cdot \mathrm{e}^{-\mathrm{r}_\gamma t}\,\mathrm{d}t = \sum_{n=0}^{N} \mu_2(\gamma, B_n)$$

By taking the limit when N goes to infinity the claim follows. It is also true that both of them are probability measure, namely $\mu_1^{\varphi,\sigma}(T) = \mu_2(\gamma,\mathbb{R}_{\geq 0}) = 1$, for every $\gamma \in T$. In particular it is the case for $\mu_1^{\varphi,\sigma}$, since there is no deadlock state in the MA. The measurability of $\mu_2(\gamma,B)$ on γ is also obvious for every $B \in \mathcal{B}(\mathbb{R}_{\geq 0})$. Therefore, both sub-measures $\mu_1^{\varphi,\sigma}$ and μ_2 satisfy the conditions of the product measure theorem. Now we can apply the theorem to build the probability measure on steps preceded by some $\varphi \in \mathbb{F}$. It is not hard to see that it then gives $\mu_\sigma(\varphi,\cdot)$, defined in Def. 3.17, as the unique probability measure on steps. □

For $\varphi \in \mathbb{F}$, the uniqueness of $\mu_\sigma(\varphi,\cdot)$ indicates that it is the only natural probability measure on measurable space $(\mathbb{S},\mathcal{S})$, in the situation when φ has just been observed.

Remark 3.19. *Different choices of sub-measures $\mu_1^{\varphi,\sigma}$ and $\mu_2(\gamma,\cdot)$ might induce different measures on steps. This means that the uniqueness of $\mu_\sigma(\varphi,\cdot)$ holds provided both of the sub-measures are given. However, since they tightly follow the MA's semantics, there is no way to have any meaningful sub-measures other than those defined in the proof of Lem. 3.18. Therefore, μ_σ is the only valid probability measure that complies with the MA's semantics.*

A notion of measurability of $\mu_\sigma(\varphi,\cdot)$ allows us to recursively employ it for building the probability measures on fragments and histories. It is described in the next lemma.

Lemma 3.20. *For every $\sigma \in \Sigma$, $S \in \mathcal{S}$ and $B \in \mathcal{B}([0,1])$, holds that $\{\varphi \in \mathbb{F}_n \mid \mu_\sigma(\varphi,S) \in B\} \in \mathcal{F}_n$.*

Proof. The claim clearly holds for $n = 0$, which also covers the case when $\varphi = \varepsilon$. Since P_{ex}^S is independent of φ, it is by Prop. 2.8 enough to show that both of the functions $f_n^\gamma, g_n^\gamma : \mathbb{F}_n \to [0,1]$ with $f_n^\gamma(\varphi) := \mathrm{d}_{n-1}^{\varphi}(\mathrm{v}_\gamma)$ and $g_n^\gamma(\varphi) := \sigma[\varphi](\mathrm{v}_\gamma)(\gamma)$ are Borel measurable for all $n > 0$ and $\gamma \in T$. To show the measurability of g_n^γ we use a similar technique as employed in the proof of Lem. 3.16. It follows from Def. 3.11 that $X_n^{B,\gamma} := \{\eta \in \mathbb{H}_n \mid \sigma(\eta)(\gamma) \in B\} \in \mathcal{H}_n$. Now let $Y_n^{B,\gamma} := \{\varphi \in \mathbb{F}_n \mid \sigma[\varphi](\mathrm{v}_\gamma)(\gamma) \in B\}$, it thus holds that $Y_n^{B,\gamma} \times \{\mathrm{v}_\gamma\} = X_n^{B,\gamma} \cap (\mathbb{F}_n \times \{\mathrm{v}_\gamma\})$. The right hand side is a measurable set, hence $Y_n^{B,\gamma}$ must be measurable, otherwise it contradicts the measurability of σ. To see the measurability of f_n^{γ}, observe that we can write $Z_n^{B,\gamma} := \{\varphi \in \mathbb{F}_n \mid f_n^{\gamma}(\varphi) \in B\}$ as

$$\mathbb{F}_{n-1} \times \Big(\big(\{\gamma' \in T_{\mathrm{M}} \mid \mathsf{d}_{\gamma'}(\mathsf{v}_\gamma) \in B\} \times \mathbb{R}_{\geq 0}\big) \cup \big(\{\gamma' \in T_{\mathrm{P}} \mid \mathsf{d}_{\gamma'}(\mathsf{v}_\gamma) \in B\} \times \{0\}\big)\Big)$$

which is a measurable rectangle. □

Now we have all ingredients to define probability measures on fragments and histories. We start with the probability measures on measurable spaces $(\mathbb{F}_n, \mathcal{F}_n)$ for all $n \in \mathbb{N}$.

Definition 3.21 (Probability measure on fragments). *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$, then the probability measure $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_n} : \mathcal{F}_n \to [0,1]$ for every $n \in \mathbb{N}$ is defined as follows. For $n = 0$, it holds that $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_0}(\{\varepsilon\}) = 1$. Otherwise:*

$$\mathfrak{p}_{v,\sigma}^{\mathbb{F}_1}(\Phi) := \mu_\sigma(\varepsilon, \{(\gamma, t) \in \Phi \mid \mathsf{v}_\gamma = v, t \in \mathbb{R}_{\geq 0}\})$$
$$\mathfrak{p}_{v,\sigma}^{\mathbb{F}_{n+1}}(\Phi) := \int_{\varphi \in \mathbb{F}_n} \mathfrak{p}_{v,\sigma}^{\mathbb{F}_n}(\mathrm{d}\varphi) \int_{\varsigma \in \mathbb{S}} \mathbb{1}_\Phi(\varphi \circ \varsigma) \cdot \mu_\sigma(\varphi, \mathrm{d}\varsigma)$$

The probability measure, as a function takes a measurable set of fragments of the respective length and determines how likely it is that they are executed by $\mathcal{M}$. The construction of the probability measure is done in a recursive manner. The case $n = 0$ is straightforward, as $\mathbb{F}_0$ only contains ε. Moreover, $\mu_\sigma(\varepsilon, \cdot)$ can be used for $n = 1$, since $\mathbb{F}_1$ and $\mathbb{S}$ coincide and also nothing is visited before the execution of each step in $\mathbb{F}_1$. The only relevant steps in this case are indeed those starting from v. Longer fragments are regarded as the extension of those that are one step shorter. That is to say, the probability measure on fragments of length n is extended into the measure on fragments of length $n + 1$ by using the probability measure on steps.

Thus far we have not established that the recursive definition given in Def. 3.21 leads to a probability measure. Once more we can employ the product measure theorem to show that it is a probability measure and indeed the only valid one that obeys the MA's semantics.

Proposition 3.22. *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$. Then, $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_n}$ is the unique probability measure on $(\mathbb{F}_n, \mathcal{F}_n)$ for every $n \in \mathbb{N}$.*

Proof. If $n = 0$, then the conclusion is obvious. For $n > 0$, the proof is done via induction. The claim for $n = 1$ follows from Lem. 3.18. The induction hypothesis yields that $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_n}$ is the unique probability measure on $(\mathbb{F}_n, \mathcal{F}_n)$. Moreover, by Lem. 3.18 for every $\varphi \in \mathbb{F}_n$, $\mu_\sigma(\varphi, \cdot)$ is the unique probability measure on $(\mathbb{S}, \mathcal{S})$. The measurability of $\mu_\sigma(\cdot, S)$ with respect to $(\mathbb{F}_n, \mathcal{F}_n)$ for all $S \in \mathcal{S}$ follows from Lem. 3.20. Therefore, both $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_n}$ and μ_σ satisfy the conditions of the product measure theorem (Thm. 2.12). Now we can apply the theorem to build the measure on $(\mathbb{F}_{n+1}, \mathcal{F}_{n+1})$ using them. It is then straightforward to show that $\mathfrak{p}_{v,\sigma}^{\mathbb{F}_{n+1}}$, as described in Def. 3.21, is the unique probability measure on the measurable space. □

By definition a history is obtained by the concatenation of a fragment and a state. Accordingly, the probability measure on fragments can be easily extended to the probability measure on histories. The uniqueness of the probability measure on fragments is then carried over to the probability measure on histories.

Proposition 3.23. *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$. Then, $\mathfrak{p}^{\mathbb{H}_n}_{v,\sigma} : \mathcal{H}_n \to [0,1]$ is the unique probability measure on $(\mathbb{H}_n, \mathcal{H}_n)$ for all $n \in \mathbb{N}$. That is, if $n = 0$ then $\mathfrak{p}^{\mathbb{H}_0}_{v,\sigma}(\{v\}) := 1$ and otherwise*

$$\mathfrak{p}^{\mathbb{H}_n}_{v,\sigma}(H) := \int_{\varphi \in \mathbb{F}} \mathfrak{p}^{\mathbb{F}_n}_{v,\sigma}(\mathrm{d}\varphi) \sum_{\{v' \mid \varphi \circ v' \in H\}} \mathrm{d}^{\varphi}_{|\varphi|-1}(v') \qquad \textit{for } H \in \mathcal{H}_n$$

Proof. The claim obviously holds for $n = 0$. For $n > 0$, put $f_n(\varphi, v) := \mathrm{d}^{\varphi}_{n-1}(v)$. It is clear that $f_n(\varphi, \cdot)$ is a probability measure for all $\varphi \in \mathbb{F}_n$. It intuitively computes the probability to land in v after execution of φ. The measurability of $f_n(\varphi, v)$ on φ for all $v \in V$ is established by the measurability of f^{γ}_n that is shown in the proof of Lem. 3.20. The claim then follows from employing product measure theorem on sub-measures $\mathfrak{p}^{\mathbb{F}_n}_{v,\sigma}$ and f_n. □

The measures on histories are used to establish the probability measure on paths. We first define a probability measure on cylinder sets, which are the main construction block of $\mathcal{P}$. As any measurable cylinder is characterised by its base, which has a finite length, the probability measure of the cylinder is determined by the probability measure of its base. It can be established that the extension leads to a unique probability measure on $(\mathbb{P}, \mathcal{P})$.

Theorem 3.24. *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$. Then there exist a unique probability measure $\mathfrak{p}_{v,\sigma} : \mathcal{P} \to [0,1]$ such that the measure of each measurable cylinder agrees with the measure of its base, i. e. for all $B_n \in \mathcal{H}_n$, it holds that $\mathfrak{p}_{v,\sigma}\big(\mathrm{Cyl}(B_n)\big) := \mathfrak{p}^{\mathbb{H}_n}_{v,\sigma}(B_n)$.*

Proof. It directly follows from Ionescu-Tulcea extension theorem. For more details see, e. g., [AD99, Thm. 2.7.2]. □

3.3 Analysis of MAs

The main purpose of this thesis is to provide sound theories and techniques for MA analysis. We have thus far proceeded one step by defining the probability measure on the set of paths. It is essential for us to have the probability measure as the basis to mathematically represent MA analysis. This is due to the fact that the analyses considered in this thesis are carried out by computing the expectation of some random variables or measurable functions defined on the set of paths. This section provides tools and methods to be used for dealing with such kinds of analyses.

We start with the definition of a general form of analysis represented by the expectation of a measurable function defined on the set of paths. Afterwards, we develop theories that aim to simplify the expectation computation. The main concept behind them is *path splitting*, which is a technique to split the analysis at a point at which some particular event happens. For instance, the reward of a path can be split into the reward before and the reward after the execution of its first transition. This helps us to express the reward computation inductively by proceeding through steps. In this section we aim to formalise such an analysis by means of the path splitting in MAs. Let us first define the expectation.

Definition 3.25 (Expectation). *Given strategy $\sigma \in \Sigma$ on MA $\mathcal{M}$ with state $v \in V$. The expectation of Borel measurable function $f : \mathbb{P} \to \overline{\mathbb{R}}$ on probability space $(\mathbb{P}, \mathcal{P}, \mathfrak{p}_{v,\sigma})$ is defined as the Lebesgue integral $\mathbb{E}_{v,\sigma}(f) := \int_{\mathbb{P}} f \, \mathrm{d}\mathfrak{p}_{v,\sigma}$*

The analysis is specified by Borel measurable function f that maps an extended real value to each path. We will see in the next chapter that the minimal and maximal values of the expectation ranging over all measurable strategies are of our particular interest.

The main idea behind the result of this section is *path splitting*. It splits a path into its prefix and suffix before and after the execution of its n-th transition, for $n > 0$. Thereafter, one can write a path $\pi \in \mathbb{P}$ as $\pi = \lceil \pi \rceil^n \circ \lfloor \pi \rfloor_n$ and then $\mathbb{P} = \mathbb{F}_n \times \mathbb{P}$. It is then possible to employ the product measure theorem to give a new representation of the probability measure on paths in terms of the probability measure on fragments of length n. It is also helpful to split the expectation computation of any measurable function according to the path splitting. It is henceforth referred to as *expectation splitting*, which is the main result of this section.

3.3.1 Possibility indicator

We have seen so far the situations in which a given fragment assumed to be executed before visiting some steps, for instance $\varphi \in \mathbb{F}$ in $\mu_\sigma(\varphi, \cdot)$ (Def. 3.17). The question arises whether it is possible at all to execute the fragment or not. It is of technical importance to know the answer to that question since some useful structures that we use are only well-defined under *possible fragments*.

Here we define an indicator to determine the possibility of a given fragment or a history. It makes use of the probability measure in a time-abstract setting, similar to the one defined for MDPs, which is naturally embedded in the probability measure in the timed setting. For our purpose we employ it independently, so as to reason about the possibility of a fragment or a history to be executed in a model under a given strategy. It thereby enables us to distinguish between *possible* and *impossible* traces.

Definition 3.26 (Possibility indicator). *Given strategy $\sigma \in \Sigma$ on MA $\mathcal{M}$, the possibility indicator $\iota_\sigma : \mathbb{F} \uplus \mathbb{H} \to [0, 1]$ under strategy σ is defined as follows. For*

$\varphi \in \mathbb{F}_n$, $\iota_\sigma(\varphi) := 1$, if $n = 0$ or 1. Otherwise, it is defined as

$$\iota_\sigma(\varphi) := \prod_{i=1}^{|\varphi|-1} \mathsf{d}_{i-1}^{\varphi}(\mathsf{v}_i^{\varphi}) \cdot \sigma(\lceil \varphi \rceil^i \circ \mathsf{v}_i^{\varphi})(\Gamma_i^{\varphi})$$

Similarly, for $\eta \in \mathbb{H}_0$, it holds $\iota_\sigma(\eta) := 1$, and $\eta \in \mathbb{H}_n$ with $n > 0$, $\iota_\sigma(\eta) := \iota_\sigma(\lceil \eta \rceil^n)\mathsf{d}_{n-1}^{\varphi}(h\downarrow)$.

The calculation of probabilities in the possibility indicator is done in a straightforward way. For each transition γ in the fragment and its successor transition γ', it multiplies the probability of landing in $\mathsf{v}_{\gamma'}$ after execution of γ with the probability of γ' being selected by σ at $\mathsf{v}_{\gamma'}$. The computation proceeds until the last pair of consecutive transitions in the fragment is processed. The possibility indicator of a history is simply that of its fragment extracted by eliminating its last state times the probability to land in the last state from its last transition.

The possibility indicator determines whether the execution of a fragment or a history is possible or impossible. In case the value is positive the fragment or the history is valid and *possible* to be executed. Conversely if the value of the indicator is zero, at least one of the transitions along the fragment or the history cannot in any way be visited. It might be due to the fact that it is not selected by the strategy or its source state is not reachable via its predecessor since in the graph representation of the MA such a predecessor does not exist. Either of the facts means that the fragment or the history is *impossible* to happen. To ease the proofs, we use the possibility indicator to avoid considering such impossible traces in our arguments. We must nevertheless establish its measurability before using it.

Lemma 3.27. *For all $n \in \mathbb{N}$ and $B \in \mathcal{B}([0,1])$, it holds that $\{\varphi \in \mathbb{F}_n \mid \iota_\sigma(\varphi) \in B\} \in \mathcal{F}_n$ and $\{\eta \in \mathbb{H}_n \mid \iota_\sigma(\eta) \in B\} \in \mathcal{H}_n$.*

Proof. We first show the claim on fragments by induction on their length. For induction bases note that the conclusion for $n = 0$ and 1 is obvious. For longer fragments it holds that

$$\iota_\sigma(\varphi) = \iota_\sigma(\lceil \varphi \rceil^{n-1}) \cdot \mathsf{d}_{n-2}^{\varphi}(\mathsf{v}_{n-1}^{\varphi}) \cdot \sigma(\lceil \varphi \rceil^{n-1} \circ \mathsf{v}_{n-1}^{\varphi})(\Gamma_{n-1}^{\varphi})$$

Therefore the claim follows from the induction hypothesis and Prop. 2.8. The measurability on fragments can be easily extended to the measurability on histories again using Prop. 2.8. □

This lemma enables us to define the subspace of possible fragments under a given strategy $\sigma \in \Sigma$. For $n \in \mathbb{N}$, let $\Upsilon_n^\sigma \subseteq \mathbb{F}_n$ be the set of all fragments in $\mathbb{F}_n$ that are possible, i. e. $\Upsilon_n^\sigma := \{\varphi \in \mathbb{F}_n \mid \iota_\sigma(\varphi) > 0\}$. The set is in fact measurable since ι_σ is measurable.

3.3.2 Expectation splitting

In this section we explain path and expectation splitting and discuss how they can possibly simplify the expectation computation for a measurable function. We proceed by introducing a probability measure that is an ingredient of path and expectation splitting. The probability measure, $\mathfrak{p}'_\sigma$, takes a possible fragment that is not null and a measurable set of paths that are executed after the fragment, and then computes the probability of the set.

Definition 3.28. *Given strategy $\sigma \in \Sigma$ on MA $\mathcal{M}$, then for $n > 0$ and $\varphi \in \Upsilon_n^\sigma$, $\mathfrak{p}'_\sigma(\varphi, \cdot) : \mathcal{P} \to [0,1]$ is defined as*

$$\mathfrak{p}'_\sigma(\varphi, \Pi) := \sum_{v \in V} \mathrm{d}_{n-1}^{\varphi}(v) \cdot \mathfrak{p}_{v,\sigma[\varphi]}(\Pi)$$

Every path $\pi \in \Pi$ is preceded by φ, i. e. event Π is studied under the condition that φ is visited before. It then makes sense to measure the probability of Π, only if φ is possible. Hence, we restrict φ to be a possible fragment. We have not yet proved that $\mathfrak{p}'_\sigma$ is a probability measure. It is shown in the next lemma.

Lemma 3.29. *Given $\sigma \in \Sigma$ on MA $\mathcal{M}$, then for every $\varphi \in \mathbb{F}$ that is* possible *and $\varphi \neq \varepsilon$, $\mathfrak{p}'_\sigma(\varphi, \cdot)$ establishes a probability measure on $(\mathbb{P}, \mathcal{P})$.*

Proof. First note that $\mathfrak{p}'_\sigma(\varphi, \cdot)$ is a measure, since it is a nonnegative function by which the empty set measures zero. Its countable additivity directly comes from that of $\mathfrak{p}_{v,\sigma[\varphi]}$. It is furthermore a probability measure, as it gives $\mathbb{P}$ the measure of one. This is due to the fact that there is not any deadlock state in $\mathcal{M}$ and also that $\mathfrak{p}_{v,\sigma'}(\mathbb{P}) = 1$ for all $v \in V$ and $\sigma' \in \Sigma$. □

In order to employ $\mathfrak{p}'_\sigma$ as a sub-measure to be used in product measure theorem or Fubini theorem, it must exhibit a notion of measurability, i. e. $\mathfrak{p}'(\cdot, \Pi)$ must be measurable for every $\Pi \in \mathbb{P}$. Since its domain is restricted to the set of possible fragments, we need to define the σ-algebra on that set. We define it using the concept of σ-algebra on a subspace by $\mathcal{Y}_n^\sigma := \{\Upsilon_n^\sigma \cap \Phi \mid \Phi \in \mathcal{F}_n\}$ for $n > 0$. This restricts each element of $\mathcal{F}_n$ to include only possible fragments. It immediately follows from the definition of $\mathcal{Y}_n^\sigma$ that reducing the domain of an arbitrary function from $\mathbb{F}_n$ to Υ_n^σ preserves its Borel measurability.

With the definition of $\mathcal{Y}_n^\sigma$ we can now discuss about the measurability of $\mathfrak{p}'_\sigma$. The intuition behind the measurability is similar to that of $\sigma[\varphi]$. Shifting the decisions of σ by fragment φ through strategy $\sigma[\varphi]$ does not violate its measurability. In a similar way it preserves the measurability of $\mathfrak{p}'_\sigma(\varphi, \Pi)$, which can be seen as the shifted version of $\mathfrak{p}_{\mathrm{v}_0^\varphi,\sigma}$. The conclusion is formally stated in the next lemma.

Lemma 3.30. *Given $\sigma \in \Sigma$ on MA $\mathcal{M}$ and $\Pi \in \mathcal{P}$, then $\mathfrak{p}'_\sigma(\cdot, \Pi) : \Upsilon_n^\sigma \to [0,1]$ is measurable on $(\Upsilon_n^\sigma, \mathcal{Y}_n^\sigma)$, for every $n > 0$.*

Proof. The proof is done by expressing $\mathfrak{p}_\sigma(\cdot, \Pi)$ in terms of well-defined arithmetic combination of measurable functions. For this we utilise the measurability of functions defined by an integral under the conditions given by Tonelli theorem. That is, consider two σ-finite measurable spaces $(\Omega_1, \mathcal{X}_1, \nu_1)$ and $(\Omega_2, \mathcal{X}_2, \nu_2)$, and let $f : \Omega_1 \times \Omega_2 \to \overline{\mathbb{R}}_{\geq 0}$ be Borel measurable on $(\Omega_1 \times \Omega_2, \mathcal{X}_1 \otimes \mathcal{X}_2)$. Then it holds that $\omega_1 \to \int_{X_2} f(\omega_1, \omega_2)\, \nu_2(\mathrm{d}\omega_2)$ is Borel measurable $(\Omega_1, \mathcal{X}_1)$, for all $X_2 \in \mathcal{X}_2$. As expected, changing the role of the measurable spaces is possible. To see the complete results delivered by the theorem see, for instance [Sch05, Thm. 13.5]. What comes in the sequel provides the construction of $\mathfrak{p}'_\sigma$ as an integral of a measurable function.

Here we make use of Dirac impulse distribution, a concept heavily applied in physics and engineering. It has the whole probability mass concentrated at one point. It is usually represented as the limit sequence of distributions such as Gaussian, Cauchy and uniform. We can, for instance, take the uniform one on the nonnegative real line:

$$\delta_k(t) := \begin{cases} k & t \in [0, \frac{1}{k}] \\ 0 & t \in (\frac{1}{k}, \infty) \end{cases}$$

for $k > 0$. It is then the case that $\lim_{k\to\infty} \delta_k(t) = \delta(t)$. The whole probability mass of $\delta(t)$ is concentrated at zero, meaning that it almost never happens to have any positive real value. It is well understood that $\lim_{k\to\infty} \int_0^\infty \delta_k(t)\,\mathrm{d}t = 1$ and for f being right continuous at 0, $\lim_{k\to\infty} \int_0^\infty f(t)\delta_k(t)\,\mathrm{d}t = f(0)$. Here, the concept of Dirac impulse distribution on the real line is carried over into fragments. That is, given a specific fragment, we introduce a distribution that concentrates its mass on that fragment and has zero mass everywhere else. We thus define $D_k : \mathbb{F}_n \times \mathbb{P} \to \mathbb{R}_{\geq 0}$ with

$$D_k(\varphi, \pi) := \prod_{i=0}^{n-1} \Big(\mathbb{1}_{\{\Gamma_i^\varphi\}}(\Gamma_i^\pi) \big(\mathbb{1}_{T_\mathrm{M}}(\Gamma_i^\pi) \delta_k(\pi\langle i\rangle - \varphi\langle i\rangle) + \mathbb{1}_{T_\mathrm{P}}(\Gamma_i^\pi) \big) \Big)$$

The function takes a fragment φ of length n and path π, and then, intuitively speaking, gives positive value only if $\lceil \pi \rceil^n$ *closely* looks like φ. By increasing k, this closedness becomes tighter and tighter, until in the limit it will give zero to all paths π whose prefix is not exactly φ. It examines the first n transition(s) of π against that of φ with their order. By using Dirac impulse distribution, it also checks that if a transition in φ is Markovian, its execution time be close to that of the respective transition in π. The Dirac impulse of fragment φ, $D(\varphi, \cdot)$, is obtained by taking the limit of sequence $\{D_k(\varphi, \cdot)\}_{k>0}$. Therefore its probability mass is concentrated on all paths that are generated by base φ. In other words, $\int_\Pi D(\varphi, \pi)\mathfrak{p}_{\nu,\sigma}(\mathrm{d}\pi) = 0$, if $\Pi \cap \mathrm{Cyl}(\varphi) = \emptyset$.

It is not hard to show that D is measurable on $(\mathbb{F}_n \times \mathbb{P}, \mathcal{F}_n \otimes \mathcal{P})$. First, consider D_k for $k > 0$; It is built up on the summations and products of indicator function and δ_k, which are obviously measurable on $(\mathbb{F}_n \times \mathbb{P}, \mathcal{F}_n \otimes \mathcal{P})$. It thus holds by

Prop. 2.8 that D_k is also measurable. Since D is the limit of sequence $\{D_k\}_{k>0}$, it is measurable by Thm. 2.9. As a result, by applying Tonelli theorem it holds that $M^{\Pi}_{v,\sigma} : \varphi \to \int_{\Pi} D(\varphi,\pi)\,\mathfrak{p}_{v,\sigma}(\mathrm{d}\pi)$ is measurable on $(\mathbb{F}_n, \mathcal{F}_n)$ for all $v \in V$ and $\Pi \in \mathcal{P}$. The same conclusion thereby holds on $(\Upsilon^{\sigma}_n, \mathcal{Y}^{\sigma}_n)$.

We now reduce the domain of D from $\mathbb{F}_n \times \mathbb{P}$ to $\mathbb{F}_n \times \mathbb{F}_n$. Observe that $D(\varphi,\pi)$ only needs $\lceil \pi \rceil^n$ for its computation. Hence, we can define $\tilde{D} : \mathbb{F}_n \times \mathbb{F}_n \to \mathbb{R}_{\geq 0}$ such that $\tilde{D}(\varphi,\varphi') := D(\varphi, \varphi' \circ \pi)$ for some arbitrary $\pi \in \mathbb{P}$. Note also that $\tilde{D}$ is measurable on $(\mathbb{F}_n \times \mathbb{F}_n, \mathcal{F}_n \otimes \mathcal{F}_n)$. This can be shown by a similar argument that is given above for the measurability of D.

Now that we have proved the measurability of $M^{\Pi}_{v,\sigma}$, we show it can be expressed by arithmetic combination of a measurable function and $\mathfrak{p}'_{\sigma}$ using Fubini theorem. More specifically, in product space $\mathbb{F}_n \times \mathbb{P}$, for some possible non-null fragment $\varphi \in \mathbb{F}_n$ and every $\varphi' \in \mathbb{F}_n$ define probability measure $\hat{\mathfrak{p}}_{\sigma,\varphi}(\varphi', \cdot) : \mathcal{P} \to [0,1]$ by $\hat{\mathfrak{p}}_{\sigma,\varphi}(\varphi', \Pi') := \mathfrak{p}'(\varphi, \Pi')$. The fact that it is a probability measure follows from Lem. 3.29. It also holds that $\varphi' \to \hat{\mathfrak{p}}_{\sigma,\varphi}(\varphi', \Pi')$ is measurable on $(\mathbb{F}_n, \mathcal{F}_n)$, for all $\Pi' \in \mathcal{P}$, simply because it is independent of φ'. Therefore it satisfies the condition to be used in Fubini theorem[4]. We aim to apply the theorem to compute $M^{\mathbb{F}_n \times \Pi}_{\mathsf{v}^{\varphi}_0,\sigma}$.

$$
\begin{aligned}
M^{\mathbb{F}_n \times \Pi}_{\mathsf{v}^{\varphi}_0,\sigma}(\varphi) &= \int_{\mathbb{F}_n \times \Pi} D(\varphi,\pi)\,\mathfrak{p}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\pi) \\
&= \int_{\mathbb{F}_n \times \Pi} \tilde{D}(\varphi, \lceil \pi \rceil^n)\,\mathfrak{p}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\pi) \\
&= \int_{\varphi' \in \mathbb{F}_n} \int_{\pi' \in \Pi} \tilde{D}(\varphi,\varphi')\,\mathfrak{p}^{\mathbb{F}_n}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\varphi')\,\hat{\mathfrak{p}}_{\sigma,\varphi}(\varphi', \mathrm{d}\pi') \\
&\qquad\qquad\qquad\qquad\qquad\qquad \big[\ast \text{ by Fubini theorem } \ast\big] \\
&= \int_{\varphi' \in \mathbb{F}_n} \int_{\pi' \in \Pi} \tilde{D}(\varphi,\varphi')\,\mathfrak{p}^{\mathbb{F}_n}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\varphi')\,\mathfrak{p}'_{\sigma}(\varphi, \mathrm{d}\pi') \\
&= \int_{\varphi' \in \mathbb{F}_n} \tilde{D}(\varphi,\varphi')\,\mathfrak{p}^{\mathbb{F}_n}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\varphi') \int_{\pi' \in \Pi} \mathfrak{p}'_{\sigma}(\varphi, \mathrm{d}\pi') \\
&= \mathfrak{p}'_{\sigma}(\varphi, \Pi) \cdot \int_{\varphi' \in \mathbb{F}_n} \tilde{D}(\varphi,\varphi')\,\mathfrak{p}^{\mathbb{F}_n}_{\mathsf{v}^{\varphi}_0,\sigma}(\mathrm{d}\varphi')
\end{aligned}
$$

Now it should become clear why $\hat{\mathfrak{p}}_{\sigma,\varphi}$ is defined independent of its second argument. It is because the value of $\tilde{D}(\varphi,\varphi')$ is zero for every $\varphi' \neq \varphi$ and thereby $\hat{\mathfrak{p}}_{\sigma,\varphi}(\varphi', \cdot)$ can give an arbitrary probability. It also follows from measurability of $\tilde{D}(\varphi,\varphi')$ on $(\mathbb{F}_n \times \mathbb{F}_n, \mathcal{F}_n \otimes \mathcal{F}_n)$ and Tonelli theorem that $\varphi \to$

[4] We actually use the classical Fubini theorem [AD99, Thm. 2.6.6], which is the corollary of Fubini theorem in the case the product measure is simply obtained by multiplying individual measures. In that case changing the order of integration is possible.

$\int_{\varphi' \in \mathbb{F}_n} \tilde{D}(\varphi, \varphi') \mathfrak{p}^{\mathbb{F}_n}_{\mathsf{v}_0^{\varphi}, \sigma}(\mathrm{d}\varphi')$ is measurable on $(\mathbb{F}_n, \mathcal{F}_n)$. Hence, the same conclusion is true on $(\Upsilon^{\sigma}_n, \mathcal{Y}^{\sigma}_n)$. Moreover the function is strictly positive for every possible $\varphi \in \mathbb{F}_n$. The claim then follows by Prop. 2.8 and the fact that $\mathfrak{p}'_{\sigma}(\varphi, \Pi)$ is a well-defined arithmetic combination of two measurable functions. □

We have now reached the moment to present the main result of this section. We have all ingredients to split expectation of a measurable function according to path splitting. That is to say, breaking a path at a specific step into its fragment up to the step and its suffix after the step. It can be useful in particular when path splitting simplifies the function, as it is the case for $D(\varphi, \pi)$ in the proof of Lem. 3.30.

Theorem 3.31 (Expectation splitting). *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$. Furthermore, assume $f : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$ is measurable with respect to $(\mathbb{P}, \mathcal{P})$. Then for $n > 0$ it holds that*

$$\begin{aligned} \mathbb{E}_{v,\sigma}(f) := \int_{\mathbb{P}} f \, \mathrm{d}\mathfrak{p}_{v,\sigma} &= \int_{\varphi \in \mathbb{F}_n} \int_{\pi \in \mathbb{P}} f(\varphi \circ \pi) \mathfrak{p}'_{\sigma}(\varphi, \mathrm{d}\pi) \mathfrak{p}^{\mathbb{F}_n}_{v,\sigma}(\mathrm{d}\varphi) \\ &= \int_{\varphi \in \mathbb{F}_n} \int_{\pi \in \mathbb{P}} f(\varphi \circ \pi) \sum_{v' \in V} \mathrm{d}^{\varphi}_{n-1}(v') \mathfrak{p}_{v', \sigma[\varphi]}(\mathrm{d}\pi) \mathfrak{p}^{\mathbb{F}_n}_{v,\sigma}(\mathrm{d}\varphi) \end{aligned}$$

Proof. It is the direct consequence of Fubini theorem. We apply the theorem on $\mathbb{P} = \mathbb{F}_n \times \mathbb{P}$ with two sub-measures $\mathfrak{p}^{\mathbb{F}_n}_{v,\sigma}$ and $\mathfrak{p}'_{\sigma}(\varphi, \cdot)$ for every $\varphi \in \mathbb{F}_n$. The crucial requirements are the fact that $\mathfrak{p}'_{\sigma}(\varphi, \cdot)$ is a probability measure for every $\varphi \in \mathbb{F}_n$, shown by Lem. 3.29, and $\mathfrak{p}'_{\sigma}(\cdot, \Pi)$ is Borel measurable for every $\Pi \in \mathcal{P}$, proven by Lem. 3.30. □

The above result is essential for proving the upper bound for the main objective considered in this thesis, namely the optimal ERR, which will be discussed in Ch. 4. It is also used to prove the fixed point characterisation of the objective, given in Ch. 5. Moreover, we can, as a result of the theorem, associate the expectation of any measurable function from an arbitrary state with the expectations in the same setting but when taking a specific transition from the state is forced. That is to say formally, $\mathbb{E}_{v,\sigma}(f)$ is a convex combination of the expectations $\mathbb{E}_{v,\sigma|_{\gamma}}(f)$ for all $\gamma \in T(v)$.

Lemma 3.32. *Let $\sigma \in \Sigma$ be a strategy on MA $\mathcal{M}$ with state $v \in V$ and assume $f : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$ is a Borel measurable function. Then, it holds that $\mathbb{E}_{v,\sigma}(f) = \sum_{\gamma \in T(v)} \sigma(v)(\gamma) \cdot \mathbb{E}_{v,\sigma|_{\gamma}}(f)$.*

Proof. The claim for $v \in V_{\mathrm{M}}$ follows from the fact that v has only one outgoing transition say $\check{\gamma}$. It thereby holds that $\sigma(v)(\check{\gamma}) = 1$ and also two strategies σ and $\sigma|_{\check{\gamma}}$ are the same. For $v \in V_{\mathrm{P}}$ we use Thm. 3.31 for $n = 1$, that is by viewing $\mathbb{P}$ as $\mathbb{S} \times \mathbb{P}$. As v is probabilistic and therefore urgent, the set of relevant steps from v is $T(v) \times \{0\}$. It is then possible to write the integral over the set by a sum. Furthermore, by putting $\varsigma = (\gamma, 0)$ for some $\gamma \in T(v)$ it is not hard to see that $\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma) = \sigma(v)(\gamma)$ and $\mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) = 1$ (by Lem. 3.21) and also $\mathfrak{p}'_\sigma(\varsigma, \mathrm{d}\pi) = \mathfrak{p}'_{\sigma|_\gamma}(\varsigma, \mathrm{d}\pi)$ (by Def. 3.28 and the fact that $\sigma[\varsigma]$ and $\sigma|_\gamma[\varsigma]$ are the same). It then gives

$$
\begin{aligned}
\mathbb{E}_{v,\sigma}(f) &= \int_{\mathbb{P}} f \,\mathrm{d}\mathfrak{p}_{v,\sigma} \\
&= \int_{\varsigma \in \mathbb{S}} \int_{\pi \in \mathbb{P}} f(\varsigma \circ \pi)\, \mathfrak{p}'_\sigma(\varsigma, \mathrm{d}\pi)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma) && \left[\ast \text{ Thm. 3.31 } \ast\right] \\
&= \sum_{\gamma \in T(v)} \int_{\pi \in \mathbb{P}} f(\gamma \xrightarrow{0} \pi)\, \mathfrak{p}'_\sigma(\gamma \xrightarrow{0}, \mathrm{d}\pi)\, \sigma(v)(\gamma) \\
&= \sum_{\gamma \in T(v)} \int_{\varsigma \in \mathbb{S}} \int_{\pi \in \mathbb{P}} f(\gamma \xrightarrow{0} \pi)\, \mathfrak{p}'_\sigma(\gamma \xrightarrow{0}, \mathrm{d}\pi)\, \sigma(v)(\gamma)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) \\
&= \sum_{\gamma \in T(v)} \sigma(v)(\gamma) \int_{\varsigma \in \mathbb{S}} \int_{\pi \in \mathbb{P}} f(\gamma \xrightarrow{0} \pi)\, \mathfrak{p}'_\sigma(\gamma \xrightarrow{0}, \mathrm{d}\pi)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) \\
&= \sum_{\gamma \in T(v)} \sigma(v)(\gamma) \int_{\varsigma \in \mathbb{S}} \int_{\pi \in \mathbb{P}} f(\gamma \xrightarrow{0} \pi)\, \mathfrak{p}'_{\sigma|_\gamma}(\gamma \xrightarrow{0}, \mathrm{d}\pi)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) \\
&= \sum_{\gamma \in T(v)} \sigma(v)(\gamma) \int_{\varsigma \in \mathbb{S}} \int_{\pi \in \mathbb{P}} f(\varsigma \circ \pi)\, \mathfrak{p}'_{\sigma|_\gamma}(\varsigma, \mathrm{d}\pi)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) \\
&= \sum_{\gamma \in T(v)} \sigma(v)(\gamma)\, \mathbb{E}_{v,\sigma|_\gamma}(f) && \left[\ast \text{ Thm. 3.31 } \ast\right]
\end{aligned}
$$

□

The lemma yields a clue about the optimal value of $\mathbb{E}_{v,\sigma}(f)$ ranging over all measurable strategies. It is indeed among $\mathbb{E}_{v,\sigma|_\gamma}(f)$ for $\gamma \in T(v)$, where a *deterministic* decision at the initial state is made. This fact is, in particular reflected in the fixed point characterisation of the optimal ERR given in Ch. 5.

As an intuitive corollary we point out the same conclusion for probability measure instead of expectation.

Corollary 3.32.1. *Given v, σ as in Lem. 3.32, then for all $\Pi \in \mathcal{P}$ it holds that* $\mathfrak{p}_{v,\sigma}(\Pi) = \sum_{\gamma \in T(v)} \sigma(v)(\gamma)\, \mathfrak{p}_{v,\sigma|_\gamma}(\Pi)$.

Proof. It is a direct consequence of Lem. 3.32 by putting $f = \mathbb{1}_\Pi$. □

3.4 Discussion

This chapter provided the mathematical basis of Markov automata and their analysis. We list the main points covered by this chapter.

(*i*) We defined Markov automaton and its compositionality in Sec. 3.1;

(*ii*) We explained the concept of *closedness* and provided the thorough semantics of *closed* MAs by means of histories and paths;

(*iii*) In Sec. 3.2, we studied the measurability of MAs, defined *generic measurable strategies* and finally established the unique probability measure on the set of paths induced by an MA;

(*iv*) For analysis of MAs, we introduced the concept of *expectation splitting* in Sec. 3.3, which splits the expectation computation for a measurable function according to a specific number of steps that has been executed.

Contributions. This chapter developed the theoretical foundations of Markov automata. We extended our previous work [HH12] on the measurability of MAs and established their unique probability measure (point (*iii*)). Moreover, we introduced the concept of expectation splitting, an abstract tool that can simplify a diverse range of analyses on MAs (point (*iv*)). Such a thorough study had not been done before for MAs.

Related works. Our development of measurability for MAs is inspired by [WJ06; Joh08]. In particular, our definition of generic measurable strategies is similar to their definition of measurable schedules for CTMDPs. They also established a probability measure for the set of paths in CTMDPs. Their work has been taken up by [Fu14b] and extended by [Neu10] to IMCs. To the best of our knowledge, expectation splitting in the general setting we considered has not been proposed before neither for MAs nor for related models (IMCs and CTMDPs). On CTMDPs however, probability measure splitting on the first step is proposed in [Fu14b, Thm. 7.1]. This is a special case of Thm. 3.31 when f is an indicator function and $n = 1$. Lem. 3.7 in [Neu10] also provides a splitting for the probability measure of CTMDPs but restricted to measurable rectangles, i. e. events of the form $\Phi \times H$ where $\Phi \in \mathcal{F}_n$ and $H \in \mathcal{H}_k$, $n, k \in \mathbb{N}$. In those, the measurability of strategies under shifting is not considered. This is an essential step that effectively makes their proof incomplete.

Future works. We did not consider different classes of strategies for MAs. Classifications of strategies for CTMDPs has been studies by [WJ06; Joh08; Neu10]. A similar classification can be applied to MAs, however in the early setting. In this chapter, we introduced the set of generic measurable strategies as the universe of MA's meaningful strategies. The question still remains open, is there a supper class of this set that still provides the unique probability measure and supports the expectation splitting offered by Thm. 3.31? The answer could be of theoretical interest.

Part II

Analysis and Characterisations

Chapter 4

Markov Reward Automata and Resource-bounded Analysis

In this chapter we introduce Markov reward automata (MRA) [Guc+14b; Hat+15; Bra+15], which as the extension of Markov automata with reward and resource structures. On MRAs, we study resource-bounded analysis and in particular we look into the optimal Expected Resource-bounded Reward (ERR). The optimal ERR is the main objective considered in this thesis. With this objective we aims to minimise or maximise the reward that can be earned by an MRA under a resource budget. It subsumes into a general category of objectives, encompassing useful analyses such as the optimal expected time-bounded reward, time- and resource-bounded reachability.

This chapter forms the theoretical basis of analysing MRAs against the optimal ERR properties. It defines relevant random variables to be used for computing the optimal ERR on MRAs. We heavily exploit sequences of functions to represent resource-bounded analysis as the limit of the expectation of a sequence of random variables. This helps us to establish useful properties of the optimal ERR by lifting characteristics of the sequence to its limit. Among them, we provide an upper bound for the optimal ERR, which is later employed as a criterion to classify MRAs. The bound is also used to prove the Lipschitz continuity of the optimal ERR.

4.1 Markov reward automaton

Often in the context of model analysis it is needed to quantify quantities such as energy consumption, profits and maintenance costs under some budget. The budget comes from *resources* such as time, money or energy. In general, resources are the quantities that are consumed in a system with a tight restriction, described as a budget. In the analyses discussed in this thesis, a bound is always imposed on resource consumption. Rewards, on the other hand, quantify the services that are offered by a system. They are therefore meant to be optimised rather than budgeted. The analyses discussed in this thesis always ask for the minimum and the maximum rewards offered by a system. Altogether, it is natural and convenient to equip models with rewards and resources. It enables us, for example, to inquire "How many tasks can be maximally completed in the system before the battery running dry?", or "What is the minimum maintenance cost of the system within one year?" In order to answer such questions we extend Markov automata with resources and rewards, which are attached to their transitions:

Definition 4.1 (Resource and reward structures)**.** *Given MA $\mathcal{M}$, a* resource function $\varrho : T \to \mathbb{R}_{\geq 0}$ *assigns a transient resource consumption rate to each transition. A* reward structure *ρ is a triple $\rho := (\rho_\mathrm{t}, \rho_\mathrm{i}, \rho_\mathrm{f})$ of functions $\rho_\mathrm{t}, \rho_\mathrm{i}, \rho_\mathrm{f} : T \to \mathbb{R}_{\geq 0}$; ρ_t is the transient reward rate, ρ_i the instantaneous reward, and ρ_f the final reward.*

For a transition $\gamma = (v, \lambda, \mu) \in T$, resources and transient rewards are granted per time unit, i. e. residing in v for t time units before taking transition γ spends $t \cdot \varrho(\gamma)$ resource units, and gains $t \cdot \rho_\mathrm{t}(\gamma)$ reward units. In contrast, the instantaneous reward $\rho_\mathrm{i}(\gamma)$ is instantly earned by executing transition γ. The final reward is granted on a state at which we run over resource budget. In this case the reward equivalent to the final reward of the next transition, which is one of the outgoing transitions of the state, is received. This construction will allow, e. g. to consider resource-bounded reachability probabilities as a special case of ERR (for more details, see Sec. 4.2.3).

We incorporate the reward and resource structures defined in Def. 4.1 in MAs to obtain *Markov reward automata*.

Definition 4.2 (Markov reward automaton)**.** Markov reward automaton (MRA) *is a triple $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ with $\mathcal{M} := (V, v_0, T)$ being a Markov automation as in Def. 3.6, ϱ and $\rho := (\rho_\mathrm{t}, \rho_\mathrm{i}, \rho_\mathrm{f})$ being its resource and respectively its reward structure as in Def. 4.1.*

Throughout this thesis, $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ is referred to as an MRA, and unless stated differently, it implicitly signifies $\mathcal{M} := (V, v_0, T)$ and $\rho := (\rho_\mathrm{t}, \rho_\mathrm{i}, \rho_\mathrm{f})$. In an MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, it is useful to distinguish transitions that consume from those that do not consume resources. It is in particular important since the transitions that do not consume any resources can be executed with no concerns about the resource budget. Hence, they can be treated differently from resource consuming transitions. The set of transitions is accordingly partitioned into *resource consuming*, denoted by T_rc, and *resource preserving* transitions, denoted by T_rp. In addition to Markovian transitions with zero resource consumption, probabilistic transitions are resource preserving as they are executed instantaneously. We can thus define $T_\mathrm{rp} := T_\mathrm{P} \cup \{\gamma \in T_\mathrm{M} \colon \varrho(\gamma) = 0\}$ and $T_\mathrm{rc} := T \setminus T_\mathrm{rp}$. The state space of a Markov automaton is partitioned accordingly. A *resource consuming* (*resource preserving*) state is the source of a resource consuming (resource preserving) transitions. The set of resource consuming and preserving transitions are denoted respectively by V_{rc} and V_{rp}. Note that $V = V_{\mathrm{rc}} \uplus V_{\mathrm{rp}}$ since there is neither a state with both resource consuming and resource preserving transitions nor a deadlock state. Moreover it obviously holds that a resource consuming state must be Markovian as well.

4.2 The optimal ERR

In this section we define the optimal ERR of an MRA using appropriate random variables that collect rewards and resources along paths induced by the MRA. We first define the functions that represent resource consumption and total reward of a fragment of a given path, and then we show that they are Borel measurable. These are described as sequences of functions indexed by the length of the corresponding fragments. We then utilise the fact that taking the limit of a sequence preserves measurability (cf. Thm. 2.9). With this we can construct a random variable for resource-bounded rewards.

4.2.1 Random reward variables

Here we propose relevant random variables for computing resources and rewards along paths. They are then employed for introducing a random variable for *resource-bounded rewards*. We consider it as the limit of a sequence of random variables. This view enables us to lift some useful properties like measurability from the sequence to its limit.

We start with resources and rewards of prefixes of a path. Given path π, the resource consumption of its prefix of length n is a function $\boldsymbol{\varrho}_n : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$, with $\boldsymbol{\varrho}_0(\pi) := 0$ and $\boldsymbol{\varrho}_n(\pi) := \sum_{i=0}^{n-1} \varrho(\Gamma_i^\pi) \cdot \pi\langle i\rangle$ for $n > 0$. It is the cumulative resource consumption of the first n transition(s) of π. The cumulative reward gained by the first n transition(s) of π is computed by $\boldsymbol{\rho}_n : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$, with $\boldsymbol{\rho}_0(\pi) := 0$ and $\boldsymbol{\rho}_n(\pi) := \sum_{i=0}^{n-1} \big(\rho_{\mathrm{t}}(\Gamma_i^\pi) \cdot \pi\langle i\rangle + \rho_{\mathrm{i}}(\Gamma_i^\pi)\big)$ for $n > 0$. The next lemma shows that both of the functions are Borel measurable.

Lemma 4.3. *$\boldsymbol{\varrho}_n$ and $\boldsymbol{\rho}_n$ are Borel measurable, for all $n \in \mathbb{N}$.*

Proof. We prove by induction on n that $\boldsymbol{\varrho}_n$ is Borel measurable. The base case, $n = 0$, is trivial. Note that $\boldsymbol{\varrho}_{n+1}(\pi) = \boldsymbol{\varrho}_n(\pi) + \varrho(\Gamma_{n-1}^\pi)\pi\langle n-1\rangle$. By induction hypothesis, $\boldsymbol{\varrho}_n$ is measurable. Therefore, by Prop. 2.8, it remains to show that $f_n(\pi) := \varrho(\Gamma_{n-1}^\pi)\pi\langle n-1\rangle$ is measurable. By Prop. 2.7 it is enough to establish, for every $b \in \mathbb{R}_{\geq 0}$, that $\{f_n \leq b\} \in \mathcal{P}$. Since $f_n(\pi')$ only depends on the n-th step of π', $\{f_n \leq b\}$ is equivalent to all paths whose n-th step is an element of $T_{\mathrm{P}} \times \{0\}$, $T_{\mathrm{rp}} \setminus T_{\mathrm{P}} \times \mathbb{R}_{\geq 0}$ or $\{(\gamma, t) \colon \gamma \in T_{\mathrm{rc}}, \varrho(\gamma) > 0, t \in [0, \frac{b}{\varrho(\gamma)})\}$. It thus constitutes a measurable base of length n whose cylinder is measurable by definition. Hence the claim follows. The proof for measurability of $\boldsymbol{\rho}_n$ is similar. □

With $\boldsymbol{\varrho}_n$ and $\boldsymbol{\rho}_n$, the resource consumption and respectively, the cumulative reward acquisition along a path is computed step by step, until the first n step(s) are visited. By imposing an extra restriction, it is possible to designate a function for computing *step- and resource-bounded rewards*. That is the reward received by a fragment of a path under a constraint on the consumption of resources. The constraint obligates the reward to be accumulated as long as the resource consumption along the path is less than or equal to some nonnegative *resource bound* or *resource budget*. It is formally given by $R_n^b : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$:

$$R_n^b(\pi) := \begin{cases} \boldsymbol{\rho}_n(\pi) & \boldsymbol{\varrho}_n(\pi) \leq b \\ \boldsymbol{\rho}_{\hat{n}-1}(\pi) + \frac{b - \boldsymbol{\varrho}_{\hat{n}-1}(\pi)}{\varrho(\Gamma_{\hat{n}}^{\pi})} \cdot \rho_{\mathrm{t}}(\Gamma_{\hat{n}}^{\pi}) + \rho_{\mathrm{f}}(\Gamma_{\hat{n}}^{\pi}) & \text{otherwise} \end{cases} \tag{4.1}$$

where $b \in \mathbb{R}_{\geq 0}$ and $\hat{n} \in \mathbb{N}$ is the length of the shortest fragment of π that consumes more than b resource units, namely $\boldsymbol{\varrho}_{\hat{n}-1}(\pi) \leq b$ and $\boldsymbol{\varrho}_{\hat{n}}(\pi) > b$. In other words, the resource consumption exceeds b while staying at the $\hat{n}$-th state of π. Therefore, the reward is gained up to residing $\frac{b - \boldsymbol{\varrho}_{\hat{n}-1}(\pi)}{\varrho(\Gamma_{\hat{n}}^{h})}$ time units in the state, and then the final reward is received. Note that $\hat{n}$ must exist and it must also hold that $\varrho(\Gamma_{\hat{n}}^{\pi}) > 0$, provided that $\boldsymbol{\varrho}_n(\pi) > b$.

Collection $\{R_n^b\}_{n \in \mathbb{N}}$ constitutes a sequence of functions converging to the random variable that we aim to propose. It is therefore necessary to establish the existence and the measurability of its limit. In the first step we show, in the next lemma, that R_n^b is measurable for every $n \in \mathbb{N}$.

Lemma 4.4. *For every $n \in \mathbb{N}$ and $b \in \mathbb{R}_{\geq 0}$, R_n^b is Borel measurable.*

Proof. Given an arbitrary $b \in \mathbb{R}_{\geq 0}$, it suffices, by Prop. 2.7, to show that $\{R_n^b \leq a\} \in \mathcal{P}$, for every $n \in \mathbb{N}$ and $a \in \mathbb{R}_{\geq 0}$. Put $g_n : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$ with $g_n(\pi) := \boldsymbol{\rho}_{\hat{n}-1}(\pi) + \frac{b - \boldsymbol{\varrho}_{\hat{n}-1}(\pi)}{\varrho(\Gamma_{\hat{n}}^{\pi})} \cdot \rho_{\mathrm{t}}(\Gamma_{\hat{n}}^{\pi}) + \rho_{\mathrm{f}}(\Gamma_{\hat{n}}^{\pi})$, as the case associated with $\boldsymbol{\varrho}_n(\pi) > b$ in Eq. (4.1). Since $\boldsymbol{\varrho}_n$ and $\boldsymbol{\rho}_n$ are measurable (Lem. 4.3) and $\varrho(\Gamma_{\hat{n}}^{\pi}) > 0$, g_n is well-defined and by Prop. 2.8 measurable. With this it holds that $\{R_n^b \leq a\} = (\{\boldsymbol{\varrho}_n \leq b\} \cap \{\boldsymbol{\rho}_n \leq a\}) \cup (\{\boldsymbol{\varrho}_n > b\} \cap \{g_n \leq a\})$. Since σ-algebras are closed under union and intersection, then $\{R_n^b \leq a\} \in \mathcal{P}$. □

The above lemma forms the basis of the random variable for computing resource-bounded rewards. Before shown to be a random variable, it is proposed in the next definition.

Definition 4.5. *For every $b \in \mathbb{R}_{\geq 0}$, $R^b : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$ is defined for every $\pi \in \mathbb{P}$ as $R^b(\pi) := \lim_{n \to \infty} R_n^b(\pi)$.*

Def. 4.5 considers R^b as the pointwise limit of the sequence $\{R_n^b\}$. We show in the next proposition that the limit exists for every $\pi \in \mathbb{P}$ and that R^b is indeed a random variable.

Proposition 4.6. *For every $b \in \mathbb{R}_{\geq 0}$, R^b is well-defined and Borel measurable.*

Proof. We show that $R^b(\pi)$ exists for every $\pi \in \mathbb{P}$. To see the existence, note that the reward and resource consumption of any transition is nonnegative. Hence, $\boldsymbol{\varrho}_n$ and $\boldsymbol{\rho}_n$ are monotonically increasing; that is to say, $\boldsymbol{\varrho}_n(\pi) \leq \boldsymbol{\varrho}_{n+1}(\pi)$ and $\boldsymbol{\rho}_n(\pi) \leq \boldsymbol{\rho}_{n+1}(\pi)$ for every $\pi \in \mathbb{P}$ and $n \in \mathbb{N}$. By Eq. (4.1), the same holds for R^b_n for every $b \in \mathbb{R}_{\geq 0}$. As a result, $\lim_{n\to\infty} R^b_n(\pi)$ exists, and thereby R^b is well-defined. To see the measurability, observe that R^b_n is measurable by Lem. 4.4 for every $n \in \mathbb{N}$, and so is its limit (Thm. 2.9). □

Measurability of R^b introduces it as a *random variable* for computing resource-bounded rewards. It delivers the integrability of R^b in the sense of Lebesgue integration. As proposed by Def. 3.25, it is thereby possible to define the expectation of R^b on probability space $(\mathbb{P}, \mathcal{P}, \mathfrak{p}_{v,\sigma})$, for $v \in V$ and $\sigma \in \Sigma$.

4.2.2 Resource-bounded analysis

This section introduces the tree of objectives we consider in this thesis. At its root is the optimal *expected resource-bounded reward* (ERR), from which other objectives branch out. We first introduce ERR from an initial state under a fixed strategy. Thereafter, we define the *optimal* ERR and show how the other objectives derive from it.

Definition 4.7 (Expected resource-bounded reward (ERR)). *Let $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ be an MRA, then its* expected resource-bounded reward *within resource bound $b \in \mathbb{R}_{\geq 0}$ from state $v \in V$ under strategy $\sigma \in \Sigma$ is the expectation of R^b on probability space $(\mathbb{P}, \mathcal{P}, \mathfrak{p}_{v,\sigma})$, i. e.*

$$\mathbf{R}(v, \sigma, b) := \mathbb{E}_{v,\sigma}(R^b) = \int_{\mathbb{P}} R^b \, \mathrm{d}\mathfrak{p}_{v,\sigma}$$

The ERR is represented for every resource bound $b \in \mathbb{R}_{\geq 0}$ as the expectation of random variable R^b with respect to an initial state under a given strategy. As it is defined in Def. 4.5 and proved in Prop. 4.6, R^b is the limit of sequence $\{R^b_n\}_{n\in\mathbb{N}}$. Due to the monotonicity of $\{R^b_n\}_{n\in\mathbb{N}}$, it is possible to establish ERR as the limit of the expectations of R^b_n.

Lemma 4.8. *Let $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ be an MRA and put $\mathbf{R}_n(v, \sigma, b) := \mathbb{E}_{v,\sigma}(R^b_n)$ for $\sigma \in \Sigma$, $v \in V$ and $b \in \mathbb{R}_{\geq 0}$. Then $\{\mathbf{R}_n(v, \sigma, b)\}_{n\in\mathbb{N}}$ forms a monotonically increasing sequence converging to $\mathbf{R}(v, \sigma, b)$.*

Proof. The claim directly follows from Thm. 2.19 (monotone convergence theorem), as R^b_n's are monotonically increasing and pointwise convergent to R^b (cf. proof of Prop. 4.6). □

The value of ERR from a fixed state and fixed resource budget may depend on the choice of strategies. In the context of model checking and verification, one is usually interested in the minimal and the maximal value that the ERR can take. That is represented by the *optimal ERR*.

Definition 4.9 (Optimal ERR)**.** *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ the minimal and the maximal ERR from state $v \in V$ within resource bound $b \in \mathbb{R}_{\geq 0}$ are respectively defined as $\underline{\mathbf{R}}(v, b) := \inf_{\sigma \in \Sigma} \mathbf{R}(v, \sigma, b)$ and $\overline{\mathbf{R}}(v, b) := \sup_{\sigma \in \Sigma} \mathbf{R}(v, \sigma, b)$.*

Thus far we have associated R^b with the sequence $\{R^b_n\}_{n \in \mathbb{N}}$ via Def. 4.5 and Prop. 4.6, and $\mathbb{E}_{v,\sigma}(R^b)$ with $\{\mathbb{E}_{v,\sigma}(R^b_n)\}_{n \in \mathbb{N}}$. It is useful to do the same for the optimal ERR. It turns out that it is not as easy as it sounds first, at least for the infimum case. To see why, we first need to define the optimal *step- and resource-bounded rewards*.

Definition 4.10. *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$* the minimal and the maximal step- and resource-bounded reward *from state $v \in V$ within resource bound $b \in \mathbb{R}_{\geq 0}$ are respectively defined as $\underline{\mathbf{R}}_n(v, b) := \inf_{\sigma \in \Sigma} \mathbf{R}_n(v, \sigma, b)$ and $\overline{\mathbf{R}}_n(v, b) := \sup_{\sigma \in \Sigma} \mathbf{R}_n(v, \sigma, b)$.*

We want to establish $\underline{\mathbf{R}}$ and $\overline{\mathbf{R}}$ as the limit of sequences $\{\underline{\mathbf{R}}_n\}_{n \in \mathbb{N}}$ and $\{\overline{\mathbf{R}}_n\}_{n \in \mathbb{N}}$, respectively. The latter follows from the fact that the limit of an increasing sequence coincides with its supremum.

Lemma 4.11. *For all $v \in V$ and $b \in \mathbb{R}_{\geq 0}$, it holds that $\lim_{n \to \infty} \overline{\mathbf{R}}_n(v, b) = \overline{\mathbf{R}}(v, b)$.*

Proof. As shown in Lem. 4.8, for every $\sigma \in \Sigma$ sequence $\{\mathbf{R}_n(v, \sigma, b)\}_{n \in \mathbb{N}}$ is monotonically increasing, and so is $\{\overline{\mathbf{R}}_n(v, b)\}_{n \in \mathbb{N}}$. It therefore holds that the limits of the sequences exist and also coincide with their supremum, i. e.

$$\lim_{n \to \infty} \mathbf{R}_n(v, \sigma, b) = \sup_{n \in \mathbb{N}} \mathbf{R}_n(v, \sigma, b) \tag{4.2}$$

$$\lim_{n \to \infty} \overline{\mathbf{R}}_n(v, b) = \sup_{n \in \mathbb{N}} \overline{\mathbf{R}}_n(v, b) \tag{4.3}$$

It then gives

$$\begin{aligned}
\lim_{n \to \infty} \overline{\mathbf{R}}_n(v, b) &= \sup_{n \in \mathbb{N}} \overline{\mathbf{R}}_n(v, b) && \big[* \text{ Eq. (4.3) } *\big] \\
&= \sup_{n \in \mathbb{N}} \sup_{\sigma \in \Sigma} \mathbf{R}_n(v, \sigma, b) && \big[* \text{ Def. 4.10 } *\big] \\
&= \sup_{\sigma \in \Sigma} \sup_{n \in \mathbb{N}} \mathbf{R}_n(v, \sigma, b) && \\
&= \sup_{\sigma \in \Sigma} \lim_{n \in \mathbb{N}} \mathbf{R}_n(v, \sigma, b) && \big[* \text{ Eq. (4.2) } *\big] \\
&= \sup_{\sigma \in \Sigma} \mathbf{R}(v, \sigma, b) && \big[* \text{ Lem. 4.8 } *\big] \\
&= \overline{\mathbf{R}}(v, b) && \big[* \text{ Def. 4.9 } *\big]
\end{aligned}$$

□

The above lemma brings a crucial result, expressing the *maximal* ERR as the limit of sequence $\{\overline{\mathbf{R}}_n\}_{n \in \mathbb{N}}$. It forms a basis for carrying some important properties of $\overline{\mathbf{R}}_n$'s such as Lipschitz continuity over to $\overline{\mathbf{R}}$. This will be discussed later in Ch. 5.

There is a subtle issue that makes the adaptation of the proof of Lem. 4.11 to the minimal case impossible. In the proof, we have changed the order of two "sup" operators, which is overall possible. However, in the minimal case, it is required to justify interchange of sup coming from the limit and inf, which is not always possible. For instance, let $f_n : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ constitutes an increasing sequence with $f_n(t) = \mathbb{1}_{[0,n]}(t)$ for $n \in \mathbb{N}$. Then, $\lim_{n\to\infty} \inf_{t\geq 0} f_n(t) = 0$, whereas $\inf_{t\geq 0} \lim_{n\to\infty} f_n(t) = 1$.

The conclusion of Lem. 4.11 can be definitely extended to the *minimal* case, nevertheless by using a more complicated proof technique. It uses a stronger notion of convergence called *uniform convergence*, which is described in Def. 2.20. For any uniformly convergent sequence of functions over a set, the limit operator over the sequence and the infimum operator over the set are commutative, as expressed in Thm. 2.21. The crucial part, however, is to show that ERR is indeed the limit of a uniformly convergent sequence.

Lemma 4.12. *For every $v \in V$ and $b \in \mathbb{R}_{\geq 0}$ the sequence $\{\mathbf{R}_n(v, \cdot, b)\}_{n\in\mathbb{N}}$ is uniformly convergent to $\mathbf{R}(v, \cdot, b)$, i. e.*

$$\lim_{n\to\infty} \sup_{\sigma\in\Sigma} \left|\mathbf{R}_n(v, \sigma, b) - \mathbf{R}(v, \sigma, b)\right| = 0$$

Proof. Proof sketch:

- we show that looking into a compact topology of strategies is enough, i. e. the strategies that regard only histories with the consumption of b resource units or less,
- we show that $\mathbf{R}_n(v, \sigma, b)$ is continuous on the topology,
- using the fact that the sequence is increasing and Dini's theorem (see e. g. [Rud76, Thm. 7.13]) the result follows.

□

Switching the order of the limit and the infimum operators is measure preserving for any uniformly convergent sequence. This is formally adapted to the minimal ERR case, as described in the next lemma.

Lemma 4.13. *For all $v \in V$ and $b \in \mathbb{R}_{\geq 0}$, it holds that* $\lim_{n\to\infty} \underline{\mathbf{R}}_n(v, b) = \underline{\mathbf{R}}(v, b)$.

Proof. It is the immediate consequence of Thm. 2.21, as $\{\mathbf{R}_n(v, \cdot\, b)\}_{n\in\mathbb{N}}$ is uniformly convergent to $\mathbf{R}(v, \cdot, b)$ by Lem. 4.12. □

4.2.3 Other types of analyses

Some other useful properties can be formulated as the optimal ERR. This is achieved by considering special reward and resource structures and/or by applying slight modifications in the model. Three important classes of properties can be considered as special cases of the optimal ERR.

Time-bounded rewards. In the *optimal time-bounded (cumulative) rewards*, time plays the role of resource. Therefore the time duration for reward collection is limited in this property. This corresponds to using resource consumption rate of one for all Markovian transitions, i. e. $\varrho := \mathbb{1}_{T_{\mathrm{M}}}$. Hence, the resource consumption become equivalent to time passage.

Resource-bounded reachability. The second class represents the *optimal resource-bounded reachability probabilities*, i. e. “what is the maximal probability to reach a set $G \subseteq V$ of states before spending b resource units?”. In order to formulate a property of this class in an MRA as the optimal ERR, the MRA is required to be slightly modified. The modification indeed preserves the value of the properties in this class. It makes each state in G absorbing, by first removing all of its outgoing transitions and then adding a Markovian self-loop with an arbitrary finite rate instead. A Markovian self-loop is a transition whose successor is its source with probability one. A special reward structure need to be applied as well. The final reward of the self-loop transitions is set to value one, namely $\rho_{\mathrm{f}}(\gamma) = 1$ if $\mathrm{v}_\gamma \in G$. All other reward values are constantly zero. As a result, each path that reaches some goal state before running out of the budget receives reward one. Thus the expected reward in that situation corresponds to the probability of reaching some state G until b resource units is spent.

Time-bounded reachability. The third class, the *optimal time-bounded reachability probabilities* is a special case of the second class, when the resource is time. It can be seen as resource-bounded reachability when the resource consumption coincides the time passage. Therefore it is formulated as the optimal ERR by applying all techniques discussed for the resource-bounded reachability as well as setting the resource consumption of all Markovian transitions to one.

4.2.4 Instantaneous resource consumption

In this thesis, we do not consider instantaneous resource consumption as it would substantially increase the hardness of the problem. To see this, we establish the complexity class of computing the optimal ERR under instantaneous resource consumption. We make use of a straightforward reduction from 0/1 knapsack problem to the optimal ERR. We formulate items in the knapsack as states of an MA and their values and weights as instantaneous rewards and resources, respectively. The result is shown in the next theorem.

Theorem 4.14. *Computation of the optimal ERR under presence of instantaneous resource consumption is NP-hard.*

Proof. We provide a reduction from the knapsack problem. The goal is to select a subset from n items, each with value x_i and weight w_i ($i = 1, \dots, n$), such that the weight of the items in the subset is at most equal to a given bound W and their value is maximal. To solve the problem, we define MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ with $\mathcal{M} = (V, v_1, T)$ such that $V := \{v_1, \ldots, v_{n+1}\}$. States $v_1, \ldots, v_n$ correspond to the respective items, whereas v_{n+1} is just a terminal state. Transitions are declared as $T := \{\gamma_i^{(0)}, \gamma_i^{(1)} \,|\, i = 1, \ldots, n\} \cup \{\gamma_{n+1}\}$ with $\gamma_i^{(0)} := \gamma_i^{(1)} := (v_i, \natural, \Delta_{v_{i+1}})$ for $i = 1, \ldots, n$ and $\gamma_{n+1} := (v_{n+1}, 1, \Delta_{v_{n+1}})$. Executing transitions $\gamma_i^{(0)}$ and $\gamma_i^{(1)}$ is equivalent to ignoring and picking item i, respectively. Transition γ_{n+1} is the Markovian self-loop of state v_{n+1} to be used for ensuring deadlock freedom. The transitions however have different resource and reward values, namely $\varrho(\gamma_i^{(1)}) = w_i$ and $\rho_{\mathrm{i}}(\gamma_i^{(1)}) = x_i$. All other rewards and resource values are zero. It is not hard to see that the knapsack problem can be solved via computing $\overline{\mathbf{R}}(v_1, W)$. □

There is another reason to avoid using the instantaneous resource consumption. The algorithmic technique for computing the optimal ERR relies on a measure preserving transformation (cf. Ch. 6) that converts resource consumption into time passage. Adding instantaneous resources would render the transformation impossible, since there is no instantaneous passage of time. Note that, in general, bringing resources into play for analysing resource-bounded properties is expensive, still for discrete-time models [AHK03], even more here so as we have to support continuous time with non-integer resources.

4.3 A bound for the optimal ERR

This section provides an upper bound for the optimal ERR of a given MRA. The bound is expressed in terms of some parameters taken from the MRA. It plays a vital role in the classification of MRAs according to the optimal ERR being finite or infinite. In addition, it will be used to prove that the optimal ERR is Lipschitz continuous with respect to resource bound and also to establish the error bound for the discretisation scheme proposed in Ch. 6.

4.3.1 Partial rewards

A reward structure consists of three different reward functions, as described in Def. 4.1. To get the resource-bounded reward of a given path, each one is applied step by step until the budget is balanced. Each step might carry a resource consuming or a resource preserving transition. We use this criterion to split the resource-bounded reward into three *partial resource-bounded rewards*. Each partial reward has its own contribution to the value of the resource-bounded reward. We study each of them separately in order to find out how large they can finally donate to the value of the optimal ERR.

For a given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ with resource bound $b \in \mathbb{R}_{\geq 0}$, we interpret random variable R^b as the sum of three orthogonal rewards, $R^b_{\mathrm{rc}}, R^b_{\mathrm{rp}}, R^b_{\mathrm{f}} : \mathbb{P} \to \overline{\mathbb{R}}_{\geq 0}$. They respectively stand for the resource-bounded reward of resource consuming transitions, that of resource preserving transitions and the resource-bounded final rewards. The first two, as the names suggest, restrict the reward to be received solely from the corresponding transitions. It can be the transient or instantaneous rewards of the transitions, but not the final rewards. The last one, R^b_{f}, only includes the final rewards. In other words, the partial rewards are all special cases of R^b, each obtained by suppressing the reward source of the others. For instance, R^b_{rc} can be obtained by setting the rewards of resource preserving transitions and the final reward of any transition to zero, i. e. $\rho_{\mathrm{t}}(\gamma) = \rho_{\mathrm{i}}(\gamma) = 0$ for all $\gamma \in T_{\mathrm{rp}}$ and $\rho_{\mathrm{f}}(\gamma') = 0$ for all $\gamma' \in T$. The others can be derived analogously. Derivation of the partial rewards from R^b indicates that they are Borel measurable.

Lemma 4.15. *For all $b \geq 0$, it holds that R^b_{rc}, R^b_{rp} and R^b_{f} are Borel measurable.*

Proof. The claim directly follows from Prop. 4.6. □

The above lemma shows that R^b_{rc}, R^b_{rp} and R^b_{f} are random variables. Moreover, it trivially holds that the random variables sum up to R^b:

$$R^b = R^b_{\mathrm{rc}} + R^b_{\mathrm{rp}} + R^b_{\mathrm{f}} \tag{4.4}$$

We are interested in the expectations of the random variables, respectively denoted by $\mathbf{R}_{\mathrm{rc}}$, $\mathbf{R}_{\mathrm{rp}}$ and $\mathbf{R}_{\mathrm{f}}$. In particular, we intend to determine the maximum possible contribution of each random variable to the value of ERR. It helps at the end to find out an upper bound of the ERR. And it can later be used to decide whether there is a states and a strategy under which the value of ERR is infinite.

For the MRA finding the maximum value of partial rewards requires determining some characteristic measures of the model. For instance, we need to know what is the maximum transient reward per time unit over all transitions of $\mathcal{M}$, or how long maximally it takes to run over the resource budget in the model. The characteristic measures depend on certain parameters of the underlying MA $\mathcal{M}$ and its resource and reward structures (ϱ and ρ) that are identified in the next two definitions.

Definition 4.16. *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, the* maximum exit rate *and the* minimum resource consumption *of resource consuming transitions are respectively defined as $\bar{\lambda} = \max_{\gamma \in T_{\mathrm{rc}}} \mathrm{r}_\gamma$ and $\underline{\varrho} = \min_{\gamma \in T_{\mathrm{rc}}} \varrho(\gamma)$.*

Def. 4.16 brings the necessary parameters for bounding the elapsed time and the expected number of Markovian jumps until the consumption of the whole resource budget in the MRA. Given a resource bound b, the former is bounded by $^b/_{\underline{\varrho}}$, which occurs while holding the transition with the minimum positive resource consumption rate until the budget is empty. The latter happens when the transition has already the highest rate, since it likely occurs more often than any other transitions in the model. The number of Markovian jumps in this case is a pure birth process or equivalently a Poisson process with rate $\bar{\lambda}$. Its expectation at time point ${}^{b}/_{\underline{\varrho}}$ is given by ${}^{\bar{\lambda}b}/_{\underline{\varrho}}$. That is indeed an upper bound for the expectation of the number of Markovian jumps in the MRA since every other transition has a smaller or equal rate and also a larger or equal resource consumption. The following definition specifies the maximum partial rewards among resource consuming transitions.

Definition 4.17. *Let* $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ *be an MRA; then among its resource consuming transitions the* maximum transient reward *is defined as* $\bar{\rho}_{\mathrm{t}} = \max_{\gamma \in T_{\mathrm{rc}}} \rho_{\mathrm{t}}(\gamma)$*, the* maximum instantaneous reward *as* $\bar{\rho}_{\mathrm{i}} = \max_{\gamma \in T_{\mathrm{rc}}} \rho_{\mathrm{i}}(\gamma)$ *and the* maximum final reward *as* $\bar{\rho}_{\mathrm{f}} = \max_{\gamma \in T_{\mathrm{rc}}} \rho_{\mathrm{f}}(\gamma)$*.*

We have now identified all parameters to be used for bounding partial rewards. The next lemma establishes upper bounds for the expectation of transient and final rewards.

Lemma 4.18. *For MRA* $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ *with resource bound* $b \in \mathbb{R}_{\geq 0}$*, it holds for all* $v \in V$*,* $\sigma \in \Sigma$ *that* $\mathbf{R}_{\mathrm{rc}}(v, \sigma, b) \leq (\bar{\rho}_{\mathrm{t}} + \bar{\lambda}\bar{\rho}_{\mathrm{i}})\frac{b}{\underline{\varrho}}$ *and* $\mathbf{R}_{\mathrm{f}}(v, \sigma, b) \leq \bar{\rho}_{\mathrm{f}}$*.*

Proof. The latter immediately follows from $R_{\mathrm{f}}(\pi) \leq \bar{\rho}_{\mathrm{f}}$ for all $\pi \in \mathbb{P}$. For the former we think of R^{b}_{rc} as the sum of two random variables obtained by projection of R^{b} onto transient and instantaneous rewards. As discussed above, the contribution of R^{b}_{rc} that comes from transient rewards is bounded above by ${}^{\bar{\rho}_{\mathrm{t}}b}/_{\underline{\varrho}}$. The expectation of the contribution from instantaneous rewards is smaller or equal to $\bar{\lambda}\bar{\rho}_{\mathrm{i}}{}^{b}/_{\underline{\varrho}}$, which proves the claim. □

It only remains to provide an upper bound for R^{b}_{rp}. Thereafter an upper bound on the ERR can be determined.

4.3.2 Bounding the optimal ERR

The previous section established bounds for R^{b}_{rc} and R^{b}_{f}, which are both finite and bounded given a model with a finite number of states and transitions and a finite resource bound. In this section we specify an upper bound for $\mathbf{R}^{b}_{\mathrm{rp}}$, which is the expectation of the resource-bounded reward of resource preserving transitions. It is possible to execute those transitions without being concerned about the resource bound as they consume no resources. Such a reward acquisition can rise to infinity. It may happen for instance when a resource preserving transition with a positive transient or instantaneous reward is visited infinitely often with positive probability.

We give another interpretation for R^{b}_{rp}. The reward from resource *preserving* transitions is collected between consecutive executions of resource *consuming* transitions. The new interpretation views the random variable as the sum of those rewards until certain *stopping time* at which the resource bound is reached. As it is well-understood in probability theory, we study the stopping time in the context of a sequence of random variables. There we make use of the theory developed in Sec. 4.2. We will see later that an upper bound for the reward earned in between two consecutive resource consuming transitions can be obtained. The bound is the adapted for R^b_{rp}.

For the sake of formality, we propose two auxiliary random variables. Firstly, random variable $\#^b\colon \mathbb{P} \to \overline{\mathbb{N}}$ is a counter of resource consuming transitions. It determines the number of resource consuming transitions executed along the path before b resource units are spent. It can be easily constructed from R^b by assigning value one to instantaneous reward of resource consuming transitions and zero to everything else. This, for example gives $\{\#^b = n\}$ as the set of all paths containing exactly n resource consuming transitions up to the point the resource consumption reaches b.

Secondly, we define sequence of random variables $A^{\emptyset}_n\colon \mathbb{P} \to \overline{\mathbb{R}}$, $n = 1, 2, \ldots$, each corresponding to the total reward of resource preserving transitions along a path up to the n-th resource consuming transition. It does not take the final reward into account. Formally it is

$$A^{\emptyset}_n(\pi) := \sum_{i=0}^{K^{\pi}_n} \big(\pi\langle i\rangle \cdot \rho_{\mathrm{t}}(\Gamma^{\pi}_i) + \rho_{\mathrm{i}}(\Gamma^{\pi}_i)\big)\, \mathbb{1}_{T_{\mathrm{rp}}}(\Gamma^{\pi}_i) \tag{4.5}$$

where K^{π}_n is the index of the n-th resource consuming transition in $\pi \in \mathbb{P}$. In other words, it is the length of the longest prefix of π that has (strictly) less than n resource consuming transitions. Therefore, K^{π}_n will be infinity if the number of resource consuming transitions along the entire path is less than n. Among all $A^{\emptyset}_n$'s, $A^{\emptyset}_1$ is of particular interest, since it can be expressed in terms of R^b_{rp} when b is carefully chosen. In fact, R^0_{rp} and $A^{\emptyset}_1$ coincide, namely $A^{\emptyset}_1(\pi) = R^0_{\mathrm{rp}}(\pi)$ for every $\pi \in \mathbb{P}$. Both compute the reward of resource preserving transitions along π up to the execution of the first resource consuming transition, at which the resource must be spend. Apart from that, it is in general possible to reconstruct R^b_{rp} using the auxiliary random variables $A^{\emptyset}_n$'s and $\#^b$ for all $b \in \mathbb{R}_{\geq 0}$. Before going into the details, we need first to prove the claim that they are measurable.

Lemma 4.19. *$\#^b$ and $A^{\emptyset}_n$ are Borel measurable for every $b \in \mathbb{R}_{\geq 0}$ and $n = 1, 2, \ldots$.*

Proof. The measurability of $\#^b$ comes from that of $\mathbf{R}^b$ by Prop. 4.6 as it is a special case of $\mathbf{R}^b$ when every resource consuming transition has instantaneous reward of one and all other rewards are zero. We prove the measurability of $A^{\emptyset}_n$ for $n = 1, 2, \ldots$ by induction on n. The conclusion for the base case follows from $A^{\emptyset}_1 = R^0_{\mathrm{rp}}$ and Lem. 4.15. It is possible to split the reward of path $\pi \in \mathbb{P}$ collected by $A^{\emptyset}_{n+1}$ into the reward up to the n-th resource consuming transition computed by $A^{\emptyset}_n$ and the reward between the n-th and the $(n+1)$-th transition specified by $A^{\emptyset}_1(\lfloor\pi\rfloor_{K^{\pi}_n+1})$, i. e. $A^{\emptyset}_{n+1}(\pi) = A^{\emptyset}_n + A^{\emptyset}_1(\lfloor\pi\rfloor_{K^{\pi}_n+1})$. By the induction hypothesis and Prop. 2.8 it only remains to show the measurability of $\pi \to A^{\emptyset}_1(\lfloor\pi\rfloor_{K^{\pi}_n+1})$, which can be proven by regarding $A^{\emptyset}_1 = R^0_{\mathrm{rp}}$. For $k \in \mathbb{N}$, let $\{K^{(\cdot)}_n = k\}$ be the set of fragments of length $k+1$ that contains n resource consuming transition(s) including its last transition. The set clearly belongs to $\mathcal{F}_{k+1}$. Then, for every

$B \in \mathcal{B}(\mathbb{R}_{\geq 0})$ the set $\{\pi \mid A_1^{\emptyset}(\lfloor \pi \rfloor_{K_n^{\pi}+1}) \in B\}$ coincides with $\bigcup_{k\in\mathbb{N}}(\{K_n^{(\cdot)} = k\} \times \{\pi \mid R_{\mathrm{rp}}^{0}(\pi) \in B\})$. The set is constructed by countable union of measurable rectangles, which proves the claim. □

Intuitively speaking, $A_n^{\emptyset}$ computes the same kind of rewards as R_{rc}^{b}, nevertheless by imposing a restriction on the number of resource consuming transitions instead of resource consumption. On the other hand, $\#^b$ can provide the right moment to stop the reward computation in terms of the number of resource consuming transitions. It can then be passed as the index of $A_n^{\emptyset}$. This composition, $A_{\#^b}^{\emptyset}$, thus collects the reward of resource preserving transitions along paths until reaching the resource consuming transition at which the resource consumption exceeds b. In fact it is nothing but R_{rp}^{b}, i. e.

$$\forall \pi \in \mathbb{P} \quad R_{\mathrm{rp}}^{b}(\pi) = A_{\#^b}^{\emptyset}(\pi) \tag{4.6}$$

In order to establish a bound for R_{rp}^{b} it is useful to express $A_{\#^b}^{\emptyset}$ in terms of $A_n^{\emptyset}$'s. We split the reward computed by $A_{\#^b}^{\emptyset}$ into the sum of the rewards received in between consecutive occurrences of resource consuming transitions. That is done by aggregating $A_1^{\emptyset}$, $A_2^{\emptyset} - A_1^{\emptyset}$, $A_3^{\emptyset} - A_2^{\emptyset}$ and so on. By using an indicator function we make sure the path is relevant, i. e. it contains a high enough number of resource consuming transitions. This is important since the evaluation of $A_{n+1}^{\emptyset}(\pi) - A_n^{\emptyset}(\pi)$ makes sense only if π contain at least n resource consuming transition(s). Altogether we can write

$$A_{\#^b}^{\emptyset} = A_1^{\emptyset} + \sum_{n=1}^{\infty}(A_{n+1}^{\emptyset} - A_n^{\emptyset})\mathbb{1}_{\{\#^b \geq n\}} \tag{4.7}$$

Our aim is to first obtain upper bounds for the expectation of the summands of Eq. (4.7) which are then combined together to provide the final bound. We start with bounding $A_1^{\emptyset}$ using the *maximal null reward* which is defined next.

Definition 4.20. *For a given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, the* maximal null reward *is defined as $r_0 = \max_{v\in V} \sup_{\sigma\in\Sigma} \mathbf{R}_{\mathrm{rp}}(v, \sigma, 0)$.*

This is simply the maximum reward from any state collected by resource preserving transitions without touching the resource budget. For any strategy $\sigma \in \Sigma$, $\mathbf{R}_{\mathrm{rp}}(v, \sigma, 0)$ is zero when v is resource consuming; otherwise it might be positive. We repeat the fact that R_{rp}^{0} in general collects the reward along the path until the first resource consuming transition is visited. At that point the reward computation stops since resources need to be spent, however the budget is empty. As R_{rp}^{0} and $A_1^{\emptyset}$ represent exactly the same random variable, maximal null reward implicitly provides an upper bound for $A_1^{\emptyset}$. All facts together brings some conclusions which is formalised in the next lemma.

Lemma 4.21. *For a given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, the following holds* *a.* $r_0 = \max_{v\in V}\sup_{\sigma\in\Sigma}\mathbb{E}_{v,\sigma}(A_1^\emptyset)$,

b. *For every* $v \in V$ *and* $\sigma\in\Sigma$, $\mathbb{E}_{v,\sigma}(A_1^\emptyset) \le r_0$,

c. *For every* $v \in V_{\mathrm{rc}}$ *and* $\sigma\in\Sigma$, $\mathbb{E}_{v,\sigma}(A_1^\emptyset) = 0$,

d. *For every* $v \in V_{\mathrm{rc}}$ *and* $\sigma\in\Sigma$, $\mathbb{E}_{v,\sigma}(A_2^\emptyset) \le r_0$.

Proof. The first three are immediate consequences of Def. 4.20 and the fact that $R_{\mathrm{rp}}^0 = A_1^\emptyset$. To prove the last one we use the technique of expectation splitting as described in Thm. 3.31 on p. 52. Starting from an arbitrary resource consuming state $v \in V_{\mathrm{rc}}$ we set strategy $\sigma\in\Sigma$ and then split the expectation of $A_2^\emptyset$ at the point v is left, i. e. by setting $n = 1$ in Thm. 3.31. Then it holds that

$$\mathbb{E}_{v,\sigma}(A_2^\emptyset) = \int_{\mathbb{P}} A_2^\emptyset \,\mathrm{d}\mathfrak{p}_{v,\sigma} = \int_{\varsigma\in\mathbb{S}}\int_{\pi\in\mathbb{P}} A_2^\emptyset(\varsigma\circ\pi)\sum_{v'\in V}\mathrm{d}_0^{\varsigma}(v')\cdot\mathfrak{p}_{v',\sigma[\varsigma]}(\mathrm{d}\pi)\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)$$

We can further simplify the equation by taking into consideration that v is resource consuming. It is also a Markovian state, which has only one outgoing transition, say $T(v) = \{\check{\gamma}\}$. It is thus only relevant to consider paths starting with step $\check{\gamma}\xrightarrow{t}$, for $t\in\mathbb{R}_{\ge 0}$; otherwise the probability measure is zero. Hence, we can restrict the first integral over $\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}$. Furthermore, it holds that $A_2^\emptyset(\varsigma\circ\pi) = A_1^\emptyset(\pi)$ for $\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}$ and $\pi\in\mathbb{P}$. It gives then

$$\begin{aligned}
\mathbb{E}_{v,\sigma}(A_2^\emptyset) &= \int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}}\int_{\pi\in\mathbb{P}} A_1^\emptyset(\pi)\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\cdot\mathfrak{p}_{v',\sigma[\varsigma]}(\mathrm{d}\pi)\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)\\
&= \int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\int_{\pi\in\mathbb{P}} A_1^\emptyset(\pi)\,\mathfrak{p}_{v',\sigma[\varsigma]}(\mathrm{d}\pi)\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)\\
&= \int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\mathbb{E}_{v',\sigma[\varsigma]}(A_1^\emptyset)\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)\\
&\le \int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,r_0\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma) && \big[*\text{ by Lem. 4.21 }*\big]\\
&= \int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}} r_0\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)\\
&= r_0\int_{\varsigma\in\{\check{\gamma}\}\times\mathbb{R}_{\ge 0}}\mathfrak{p}^{\mathbb{S}}_{v,\sigma}(\mathrm{d}\varsigma)\\
&= r_0
\end{aligned}$$

□

Thus far a general upper bound is established for the expectation of $A_1^\emptyset$. And the bound is carried over into the expectation of $A_2^\emptyset$ starting from resource consuming states. We have now all ingredients to extend the bound to $\mathbf{R}_{\mathrm{rp}}^b$ via random variable $A_{\#b}^\emptyset$ for an arbitrary nonnegative resource bound b.

Lemma 4.22. *Given an MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ and let $b \geq 0$ be a resource bound. Then for every $\sigma \in \Sigma$, it holds that $\mathbf{R}_{\text{rp}}(v, \sigma, b) \leq \frac{\bar{\lambda} b}{\underline{\varrho}} r_0$ if $v \in V_{\text{rc}}$, and $\mathbf{R}_{\text{rp}}(v, \sigma, b) \leq r_0 + \frac{\bar{\lambda} b}{\underline{\varrho}} r_0$ otherwise.*

Proof. The proof proceeds in two steps. We first relate $\mathbf{R}_{\text{rp}}$ to the reward collected in between consecutive executions of resource consuming transitions using Eqs. (4.6) and (4.7):

$$\begin{aligned}
\mathbf{R}_{\text{rp}}(v, \sigma, b) &= \mathbb{E}_{v,\sigma}\big(R^b_{\text{rp}}\big) \\
&= \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_{\#^b}\big) && \big[* \text{ by Eq. (4.6) } *\big] \\
&= \mathbb{E}_{v,\sigma}\Big(A^{\emptyset}_1 + \sum_{n=1}^{\infty}(A^{\emptyset}_{n+1} - A^{\emptyset}_n)\mathbb{1}_{\{\#^b \geq n\}}\Big) && \big[* \text{ by Eq. (4.7) } *\big] \\
&= \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_1\big) + \sum_{n=1}^{\infty}\mathbb{E}_{v,\sigma}\Big((A^{\emptyset}_{n+1} - A^{\emptyset}_n)\mathbb{1}_{\{\#^b \geq n\}}\Big) && (4.8)
\end{aligned}$$

where the last equation comes from the linearity of expectation. The second step is crucial to establish an upper bound for the last term of Eq. (4.8). Formally, we would like to determine an upper bound for

$$\begin{aligned}
\mathbb{E}_{v,\sigma}\Big((A^{\emptyset}_{n+1} - A^{\emptyset}_n)\mathbb{1}_{\{\#^b \geq n\}}\Big) &= \int_{\mathbb{P}} (A^{\emptyset}_{n+1} - A^{\emptyset}_n)\mathbb{1}_{\{\#^b \geq n\}} \,\mathrm{d}\mathfrak{p}_{v,\sigma} \\
&= \int_{\{\#^b \geq n\}} (A^{\emptyset}_{n+1} - A^{\emptyset}_n) \,\mathrm{d}\mathfrak{p}_{v,\sigma} && (4.9)
\end{aligned}$$

for every $n = 1, 2, \ldots$. We proceed by partitioning $\{\#^b \geq n\}$ according to the length of prefixes that contain exactly n resource consuming transition(s). Let $\overline{C}^k_n$ be the set of all paths whose shortest prefix with n resource consuming transition(s) has length k. The set of such prefixes, namely the fragments of length k containing n resource consuming transition(s) including its last transition, is denoted by C^k_n. It obviously holds that $\overline{C}^k_n = \text{Cyl}(C^k_n)$ and $\lceil \pi \rceil^k \in C^k_n$ for $\pi \in \overline{C}^k_n$. It is not hard to see that $\overline{C}^k_n$ is empty for $k \leq n$, also $\overline{C}^k_n$'s are pairwise disjoint for $k \geq n$. Therefore, they are used to partition $\{\#^b \geq n\}$. Let $\overline{\Gamma}^b_{n,k} := \overline{C}^k_n \cap \{\#^b \geq n\}$, then it holds that $\{\#^b \geq n\} = \biguplus_{k=n}^{\infty} \overline{\Gamma}^b_{n,k}$. With this we rewrite Eq. (4.9) as

$$\begin{aligned}
\mathbb{E}_{v,\sigma}\Big((A^{\emptyset}_{n+1} - A^{\emptyset}_n)\mathbb{1}_{\{\#^b \geq n\}}\Big) &= \int_{\{\#^b \geq n\}} (A^{\emptyset}_{n+1} - A^{\emptyset}_n) \,\mathrm{d}\mathfrak{p}_{v,\sigma} \\
&= \int_{\biguplus_{k=n}^{\infty} \overline{\Gamma}^b_{n,k}} (A^{\emptyset}_{n+1} - A^{\emptyset}_n) \,\mathrm{d}\mathfrak{p}_{v,\sigma} \\
&\overset{(\dagger)}{=} \sum_{k=n}^{\infty} \int_{\overline{\Gamma}^b_{n,k}} (A^{\emptyset}_{n+1} - A^{\emptyset}_n) \,\mathrm{d}\mathfrak{p}_{v,\sigma}
\end{aligned}$$

$$= \sum_{k=n}^{\infty} \int_{\mathbb{P}} (A^{\emptyset}_{n+1} - A^{\emptyset}_{n}) \mathbb{1}_{\overline{\Gamma}^b_{n,k}} \, \mathrm{d}\mathfrak{p}_{v,\sigma} \tag{4.10}$$

where (†) derives from the fact that Lebesgue integral is countably additive (see e.g. [AD99, Thm. 1.6.1]). Now we apply expectation splitting, as described in Thm. 3.31, into Eq. (4.10) by taking the prefixes of length k. Before going into details we specify some facts about the splitting. Moreover, similar to $\overline{C}^k_n$, we write $\overline{\Gamma}^b_{n,k}$ as the cylinder of a measurable base. Take any $\pi \in \overline{\Gamma}^b_{n,k}$, then, by definition, it holds that $\lceil \pi \rceil^k \in C^k_n$ and $\#^b(\pi) \geq n$. Since $\lceil \pi \rceil^k$ already contains n resource consuming transition(s), its resource consumption is less than b, i. e. $\boldsymbol{\varrho}_k(\pi) \leq b$. Hence, it holds for every $\pi' \in \mathrm{Cyl}(\lceil \pi \rceil^k)$ that $\boldsymbol{\varrho}_k(\pi') \leq b$ and thereby $\pi' \in \overline{\Gamma}^b_{n,k}$. To summarise, let $\Gamma^b_{n,k}$ be the set of fragment of length k such that $\overline{\Gamma}^b_{n,k} = \mathrm{Cyl}(\Gamma^b_{n,k})$[1], then $\pi \in \overline{\Gamma}^b_{n,k}$ iff $\lceil \pi \rceil^k \in \Gamma^b_{n,k}$. We accordingly simplify $A^{\emptyset}_{n+1}(\pi)$ for every $\pi \in \overline{\Gamma}^b_{n,k}$ as

$$A^{\emptyset}_{n+1}(\pi) = A^{\emptyset}_{n}(\pi) + A^{\emptyset}_{1}(\lfloor \pi \rfloor_k) \tag{4.11}$$

since $\lceil \pi \rceil^k$ has n resource consuming transition(s) including its last one. Therefore

$$\mathbb{E}_{v,\sigma}\Big((A^{\emptyset}_{n+1} - A^{\emptyset}_{n}) \mathbb{1}_{\{\#^b \geq n\}}\Big) = \sum_{k=n}^{\infty} \int_{\mathbb{P}} (A^{\emptyset}_{n+1} - A^{\emptyset}_{n}) \mathbb{1}_{\overline{\Gamma}^b_{n,k}} \, \mathrm{d}\mathfrak{p}_{v,\sigma} \quad \big[* \text{ by Eq. (4.10) } *\big]$$

$$= \sum_{k=n}^{\infty} \int_{\varphi \in \mathbb{F}_k} \int_{\pi \in \mathbb{P}} \Big(\big(A^{\emptyset}_{n+1}(\varphi \circ \pi) - A^{\emptyset}_{n}(\varphi \circ \pi)\big) \mathbb{1}_{\overline{\Gamma}^b_{n,k}}(\varphi \circ \pi)\Big)$$

$$\times \sum_{v' \in V} \mathrm{d}^{\varphi}_{k-1}(v') \, \mathfrak{p}_{v',\sigma[\varphi]}(\mathrm{d}\pi) \, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi) \quad \big[* \text{ by Thm. 3.31 } *\big]$$

$$= \sum_{k=n}^{\infty} \int_{\varphi \in \Gamma^b_{n,k}} \int_{\pi \in \mathbb{P}} \big(A^{\emptyset}_{n+1}(\varphi \circ \pi) - A^{\emptyset}_{n}(\varphi \circ \pi)\big)$$

$$\sum_{v' \in V} \mathrm{d}^{\varphi}_{k-1}(v') \, \mathfrak{p}_{v',\sigma[\varphi]}(\mathrm{d}\pi) \, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi)$$

$$= \sum_{k=n}^{\infty} \int_{\varphi \in \Gamma^b_{n,k}} \int_{\pi \in \mathbb{P}} A^{\emptyset}_{1}(\pi) \sum_{v' \in V} \mathrm{d}^{\varphi}_{k-1}(v') \, \mathfrak{p}_{v',\sigma[\varphi]}(\mathrm{d}\pi) \, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi)$$

$$\big[* \text{ by Eq. (4.11) } *\big]$$

$$= \sum_{k=n}^{\infty} \int_{\varphi \in \Gamma^b_{n,k}} \sum_{v' \in V} \int_{\pi \in \mathbb{P}} A^{\emptyset}_{1}(\pi) \mathrm{d}^{\varphi}_{k-1}(v') \, \mathfrak{p}_{v',\sigma[\varphi]}(\mathrm{d}\pi) \, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi)$$

$$= \sum_{k=n}^{\infty} \int_{\varphi \in \Gamma^b_{n,k}} \sum_{v' \in V} \mathrm{d}^{\varphi}_{k-1}(v') \, \mathbb{E}_{v',\sigma[\varphi]}(A^{\emptyset}_{1}(\pi)) \, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi)$$

[1] It is important to note that projection does not in general preserve measurability of sets. However, it is the case here, since the set is actually a cylinder.

$$\begin{aligned}
&\leq \sum_{k=n}^{\infty} \int_{\varphi\in\Gamma^b_{n,k}} \sum_{v'\in V} \mathrm{d}^{\varphi}_{k-1}(v')\, r_0\, \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi) && \left[\ast \text{ by Lem. 4.21 } \ast\right]\\
&= r_0 \cdot \sum_{k=n}^{\infty} \int_{\varphi\in\Gamma^b_{n,k}} \mathfrak{p}_{v,\sigma}(\mathrm{d}\varphi)\\
&= r_0 \cdot \sum_{k=n}^{\infty} \mathfrak{p}^{\mathbb{F}_k}_{v,\sigma}(\Gamma^b_{n,k})\\
&= r_0 \cdot \sum_{k=n}^{\infty} \mathfrak{p}_{v,\sigma}(\overline{\Gamma}^b_{n,k}) && \left[\ast \text{ by Thm. 3.24 } \ast\right]\\
&= r_0 \cdot \mathfrak{p}_{v,\sigma}\Big(\biguplus_{k=n}^{\infty} \overline{\Gamma}^b_{n,k}\Big) && \left[\ast \text{ by countable additivity of } \mathfrak{p}_{v,\sigma} \ast\right]\\
&= r_0 \cdot \mathfrak{p}_{v,\sigma}(\#^b \geq n) && (4.12)
\end{aligned}$$

We proceed by putting Eq. (4.12) into (4.8), which finally gives the desired upper bound.

$$\begin{aligned}
\mathbf{R}_{\mathrm{rp}}(v,\sigma,b) &= \mathbb{E}_{v,\sigma}\big(R^b_{\mathrm{rp}}\big)\\
&= \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_{\#^b}\big) && \left[\ast \text{ Eq. (4.6) } \ast\right]\\
&\leq \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_{1}\big) + r_0 \cdot \sum_{n=1}^{\infty} \mathfrak{p}_{v,\sigma}(\#^b \geq n) && \left[\ast \text{ Eq. (4.12) } \ast\right]\\
&= \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_{1}\big) + r_0 \cdot \mathbb{E}_{v,\sigma}(\#^b) && \left[\ast \text{ Lem. 2.16 } \ast\right]\\
&= \mathbb{E}_{v,\sigma}\big(A^{\emptyset}_{1}\big) + r_0 \cdot \frac{\bar{\lambda} b}{\underline{\varrho}} && \left[\ast \text{ Lem. 4.18 } \ast\right]
\end{aligned}$$

The last equation is derived from Lem. 4.18 by viewing $\#^b$ as a special case of R^b_{rc} when the transient and instantaneous reward of all resource consuming transitions are respectively zero and one. The conclusion therefore follows from Lem. 4.21 by the fact that $\mathbb{E}_{v,\sigma}(A^{\emptyset}_1) = 0$ for $v \in T_{\mathrm{rc}}$, and $\mathbb{E}_{v,\sigma}(A^{\emptyset}_1) \leq r_0$ otherwise. □

The proved upper bound for $\mathbf{R}_{\mathrm{rp}}$ together with the result of Lem. 4.18 provide the necessary ingredients to bound ERR. We are now ready to represent the main result of this section:

Proposition 4.23. *Given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ with resource bound $b \in \mathbb{R}_{\geq 0}$. Then for every $\sigma \in \Sigma$, it holds that $\mathbf{R}(v,\sigma,b) \leq \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{i}} + r_0\big) + \bar{\rho}_{\mathrm{f}}$ for $v \in V_{\mathrm{rc}}$ and $\mathbf{R}(v,\sigma,b) \leq r_0 + \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{i}} + r_0\big) + \bar{\rho}_{\mathrm{f}}$ otherwise.*

Proof. We view random variable R^b as the sum of three partial rewards as described in Eq. (4.4). We then apply the upper bound for the expectation of the random variables. Assume for the moment that $v \in V_{\mathrm{rc}}$, then

$$\begin{aligned}
\mathbf{R}(v,\sigma,b) &= \mathbf{R}_{\mathrm{rc}}(v,\sigma,b) + \mathbf{R}_{\mathrm{rp}}(v,\sigma,b) + \mathbf{R}_{\mathrm{f}}(v,\sigma,b) \\
&\qquad\qquad \big[\ast \text{ Eq. (4.4) and linearity of expectation } \ast\big] \\
&\le \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}} \bar{\rho}_{\mathrm{i}} + r_0 + \frac{\bar{\lambda} b}{\underline{\varrho}} r_0 + \bar{\rho}_{\mathrm{f}} \qquad \big[\ast \text{ Lem. 4.18 and 4.22 } \ast\big] \\
&= r_0 + \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}} \big(\bar{\rho}_{\mathrm{i}} + r_0\big) + \bar{\rho}_{\mathrm{f}}
\end{aligned}$$

The conclusion for $v \in V_{\mathrm{rp}}$ is achieved by dropping r_0 from the above bound justified by Lem. 4.22. □

The bound can be carried over into the optimal ERR.

Corollary 4.23.1. *Assume the hypothesis of Prop. 4.23 then it holds that* $\underline{\mathbf{R}}(v,b) \le \overline{\mathbf{R}}(v,b) \le \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{i}} + r_0\big) + \bar{\rho}_{\mathrm{f}}$ *for* $v \in V_{\mathrm{rc}}$ *and* $\underline{\mathbf{R}}(v,b) \le \overline{\mathbf{R}}(v,b) \le r_0 + \frac{\bar{\rho}_{\mathrm{t}} b}{\underline{\varrho}} + \frac{\bar{\lambda} b}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{i}} + r_0\big) + \bar{\rho}_{\mathrm{f}}$ *otherwise.*

Apart from proving an upper bound for the maximal ERR, Cor. 4.23.1 clarifies how the quantity can be unbounded. Indeed the only part that can be divergent is the one coming from resource preserving transitions, namely $\mathbf{R}_{\mathrm{rp}}$. It is used as a criterion to classify MRAs.

4.4 MRA classification

This section is dedicated to the classification of MRAs according to the value of the maximal ERR. We employ the result of the previous section to distinguish MRAs whose maximal ERR is finite from those whose maximal ERR is infinite. Prop. 4.23 and Cor. 4.23.1 provide the criterion for drawing the distinction. To ensure the finiteness of (optimal) ERR it is enough to check the value of maximal null reward being finite. Since the state and transition spaces of the model are finite, by Def. 4.20 the maximal null reward from all states are finite. Accordingly, we classify states of an MRA and also MRAs themselves.

Definition 4.24. *In a given MRA* $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$*, state* v *is* null reward convergent (NRC) *iff* $\sup_{\sigma\in\Sigma} \mathbf{R}_{\mathrm{rp}}(v,\sigma,0) < \infty$*; conversely, it is* null reward divergent (NRD)*. An MRA is NRD if it contains an NRD states, otherwise it is NRC.*

The classification helps us further to judge whether the ERR is finite or infinite. Following from Def. 4.24 and Cor. 4.23.1, being NRC is conducive to finiteness of the (optimal) ERR in spite of how big the resource budget is. It is an interesting and useful result to be able to determine the finiteness of the maximal ERR under any resource budget by examining only maximal null rewards from all states. It again clarifies why maximal null rewards are the only reason for the divergence of (optimal) ERR. The following lemma represents it formally.

Lemma 4.25. *Given an NRC MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, then $\overline{\mathbf{R}}(v, b)$ is finite for all $v \in V$ and $b \in \mathbb{R}_{\geq 0}$.*

Proof. By Def. 4.20 and 4.24, r_0 for an NRC MRA is finite. The claim then follows from Cor. 4.23.1. □

NRC models exhibit distinctive features. In particular, their optimal ERR as a function of resource budget is Lipschitz continuous. This will be discussed in detail in Ch. 5. We will in addition present their corresponding Lipschitz constant by means of Cor. 4.23.1. The contrapositive of the above lemma is also important. It states that if the maximal ERR under some budget $b \in \mathbb{R}_{\geq 0}$ starting from some state v is infinite, then the MRA is NRD. Furthermore, it is possible to develop an approximation scheme for computing the optimal ERR of NRC models with a strict error bound. This will be proposed in Ch. 6.

4.5 Discussion

This chapter defined Markov reward automata and provided the basis of their analysis against the optimal ERR. Here comes the list of the main points discussed in this chapter.

(*i*) We defined *Markov reward automata* in Sec. 4.1 by combining Markov automata with resources and rewards;

(*ii*) We introduced the optimal ERR in Sec. 4.2 and showed that it is the limit of the sequence of optimal *expected step- and resource-bounded rewards*;

(*iii*) We established an upper bound for the optimal ERR by considering resource preserving, resource consuming and final rewards separately;

(*iv*) We classified MRAs as either *null reward convergent* or *null reward divergent* models according to their optimal ERR being finite or infinite.

Contributions. To introduce Markov reward automata and their analysis under resource constraints, we extended and generalised our previous works [HH12; Guc+13; Guc+14b; Guc+14a; Bra+15; Hat+15] from *timed reachability* analysis of MAs to resource-bounded reward analysis of MRAs. Such a broad analysis (on finite horizon) has never been done before for neither MRAs nor MAs and similar models including IMCs and CTMDPs. We also showed that the *minimal* case is technically more difficult than the *maximal* case. While most of the previous works restricted their analysis to the maximal case, we developed our theory for both (point (*ii*)). Establishing an upper bound for the optimal ERR of MRAs is the other contribution of this chapter (point (*iii*)). To this end we utilised the rigorous technique of expectation splitting developed in Ch. 3. This enabled us to, depending on the value of the optimal ERR being finite or infinite, classified MRAs into *null reward convergent* and *null reward divergent*, respectively (point (*iv*)).

Related works. One of the basic works on the analysis of resource bounded properties for nondeterministic and stochastic systems is done by [Bai+08]. It considers the problem of resource-bounded reachability of *uniform* CTMDPs under time abstract strategies. Here, we have generalised the result to richer classes of properties, strategies and models. We also relaxed the constraints on positive resource consumption by supporting *resource preserving* transitions. Resource-bounded reachability for CTMDPs is also studied by [Fu14a; Fu14b]. Comparing to that work, we generalised reachability to rewards on a different model, namely MRAs. Other differences between the two methods and a more complete list of related works are discussed in Sec. 5.4 and 6.5.

Future works. Expectation splitting in principle enables us to simplify a broad range of analyses in MRAs. It is therefore worthwhile to investigate more general analysis classes and to study their properties using the expectation splitting technique. Discounted and time-inhomogeneous rewards are notable examples. Furthermore, a thorough study of *null reward divergence* and its connection to *timelock* in MRAs has not been done yet. We strongly believe that there is a relation between timelock and the uniqueness of the optimal ERR as the fixed point of the characterisation that will be given in Ch. 5. Proving the existence of reward divergent strategy is also important, so as to develop an algorithm to detect null reward divergent models.

Chapter 5

Characteristics of the optimal ERR

This chapter demonstrates some important properties of the optimal expected resource-bounded reward (ERR). We characterise the objective as a fixed point of a Volterra integral equation. This present an opportunity to develop an algorithm for computing the optimal ERR. More specifically, it provides the basis for the discretisation scheme and its error analysis that will be discussed in the next chapter. Furthermore, we study the continuity of the optimal ERR. We exploit the fixed point characterisation to show the optimal ERR of MRAs that are null reward convergent (NRC) is Lipschitz continuous. Moreover, we present the corresponding Lipschitz constant using the upper bound of the optimal ERR, proposed in the previous chapter. At the end of this chapter, we look into the derivative of the optimal ERR with respect to the resource budget and show that it exists almost everywhere. Moreover, we discuss where the derivative may not exist.

5.1 A fixed point characterisation

The aim of this section is to provide a characterisation of the optimal ERR as a fixed point. We approach the problem step-by-step and first recap on the fact that the optimal ERR is the limit of a sequence of functionals. More precisely, the sequence is constituted by the optimal expected step- and resource-bounded rewards as described in Def. 4.10. We have previously shown in Lem. 4.11 and 4.13 that the limit converges to the optimal ERR. As explained in the previous chapter, the infimum case appears with a technical difficulty that complicates the proof. Throughout this chapter, we take it into consideration and explain different situations arising by both infimum and supremum cases.

Here we recap the definition of the optimal expected n-step resource-bounded rewards. Given an MRA with state v and a resource budget $b \in \mathbb{R}_{\geq 0}$. Then the minimal and maximal n-step and resource-bounded rewards are respectively defined for $n \in \mathbb{N}$ as:

$$\underline{\mathbf{R}}_n(v,b) := \inf_{\sigma\in\Sigma} \mathbf{R}_n(v,\sigma,b) = \inf_{\sigma\in\Sigma} \mathbb{E}_{v,\sigma}(R_n^b) \tag{5.1}$$

$$\overline{\mathbf{R}}_n(v,b) := \sup_{\sigma\in\Sigma} \mathbf{R}_n(v,\sigma,b) = \sup_{\sigma\in\Sigma} \mathbb{E}_{v,\sigma}(R_n^b) \tag{5.2}$$

where R_n^b is the random variable for n-step resource-bounded rewards as defined in Eq. (4.1) on p. 60. We may use $\overline{\underline{\mathbf{R}}}_n$ to denote either of the minimal or the maximal value and similarly for $\overline{\underline{\mathbf{R}}}$. Here, we provide a characterisation of $\overline{\underline{\mathbf{R}}}_n$ as a recursive integral equation. We will see later that its fixed point is the solution of the induced Volterra integral equation.

Lemma 5.1. *Let $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ be an MRA and $b \in \mathbb{R}_{\geq 0}$ be a resource bound. Then, for $n \in \mathbb{N}$ it holds that*

$$\begin{aligned}\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = &\int_0^{b/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\ &+ \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\end{aligned} \tag{5.3a}$$

if $v \in V_{\mathrm{rc}}$ with $T(v) = \{\check{\gamma}\}$. For $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$:

$$\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b) \tag{5.3b}$$

And finally for $v \in V_{\mathrm{P}}$:

$$\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = \operatorname*{mix}_{\gamma\in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V} \mathrm{d}_{\gamma}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b)\Big) \tag{5.3c}$$

where mix *is* max *if* $\overline{\mathbf{R}}$ *is meant and* min *otherwise. In the base case,* $\overline{\underline{\mathbf{R}}}_0$ *is constantly zero.*

Proof. We first show that to optimise ERR it suffices to only considers strategies that decide deterministically for the initial state. For that we use the result of Lem. 3.32 which represents the expectation of a function as a convex combination of several expectations. Obviously, the maximum of those expectations is superior to any of their convex combination. The same holds for the minimum. We employ the fact here, first in the maximum setting.

$$\begin{aligned}\overline{\mathbf{R}}_n(v,b) &= \sup_{\sigma\in\Sigma} \mathbb{E}_{v,\sigma}(R_n^b) && \big[\ast\ \text{Eq. (5.2)}\ \ast\big] \\ &= \sup_{\sigma\in\Sigma} \sum_{\gamma\in T(v)} \sigma(v)(\gamma)\,\mathbb{E}_{v,\sigma|_\gamma}(R_n^b) && \big[\ast\ \text{Lem. 3.32}\ \ast\big] \\ &= \sup_{\sigma\in\Sigma} \max_{\gamma\in T(v)} \mathbb{E}_{v,\sigma|_\gamma}(R_n^b) \\ &= \max_{\gamma\in T(v)} \sup_{\sigma\in\Sigma} \mathbb{E}_{v,\sigma|_\gamma}(R_n^b) \\ &= \max_{\gamma\in T(v)} \sup_{\sigma\in\Sigma} \mathbf{R}_n(v,\sigma|_\gamma,b) && (5.4)\end{aligned}$$

It can be easily shown that the same holds in the minimum case:

$$\overline{\mathbf{R}}_n(v,b) = \min_{\gamma\in T(v)} \inf_{\sigma\in\Sigma} \mathbf{R}_n(v,\sigma|_\gamma,b) \tag{5.5}$$

We have thus far presented that for the optimal ERR up to n step(s) it is enough to optimise over finitely many choices, each corresponding to a transition that can be taken at the initial state. Afterwards, we elaborate on the choices for different types of states. We distinguish between the cases spelled out in the theorem.

For $v \in V_{\mathrm{rc}}$ there is only one outgoing transition, say $\check{\gamma}$, which must be Markovian. Therefore, we can skip max operator of Eq. (5.4) (min operator of Eq. (5.5)) and write

$$\overline{\underline{\mathbf{R}}}_n(v,b) = \operatorname*{opt}_{\sigma\in\Sigma} \mathbf{R}_n(v,\sigma|_{\check{\gamma}},b) \tag{5.6}$$

where opt is either inf or sup respecting the choice of $\overline{\underline{\mathbf{R}}}_n$, which is either $\underline{\mathbf{R}}_n$ or $\overline{\mathbf{R}}_n$. At this time we use expectation splitting given by Thm. 3.31 to simplify $\mathbf{R}_{n+1}(v,\sigma|_{\check{\gamma}},b)$. We split the expectation at the first step; then it holds that

$$\mathbf{R}_n(v,\sigma|_\gamma,b) = \int_{\varsigma\in\mathbb{S}} \int_{\pi\in\mathbb{P}} R_n^b(\varsigma\circ\pi) \sum_{v'\in V} \mathrm{d}_0^{\varsigma}(v')\, \mathfrak{p}_{v',\sigma|_{\check{\gamma}}[\varsigma]}(\mathrm{d}\pi)\, \mathfrak{p}^{\mathbb{S}}_{v,\sigma|_{\check{\gamma}}}(\mathrm{d}\varsigma) \tag{5.7}$$

We can further simplify the equation by using the fact that $\check{\gamma}$ is the only outgoing transition of v and thereby σ and $\sigma|_{\check{\gamma}}$ are the same. It is then only relevant to consider paths starting with step $\varsigma \in \{\check{\gamma}\} \times \mathbb{R}_{\geq 0}$. Moreover, by Def. 3.21 it holds for every $B \in \mathcal{B}(\mathbb{R}_{\geq 0})$ that

$$\mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\{\check{\gamma}\}\times B) = \int_B \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t}\, \mathrm{d}t \tag{5.8}$$

With Eq. (5.8) one can rewrite Eq. (5.7) in order to adapt the first integral to be taken over the nonnegative real line. That is indeed possible since, as mentioned before, there is only one relevant choice for the transition of the first step to be executed at any time point on the time horizon. Therefore we can write

$$\mathbf{R}_n(v,\sigma|_{\check{\gamma}},b) = \int_0^\infty \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \int_{\pi\in\mathbb{P}} R_n^b(\varsigma\circ\pi) \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\, \mathfrak{p}_{v',\sigma[\varsigma]}(\mathrm{d}\pi)\, \mathrm{d}t \tag{5.9}$$

Note that Eqs. (5.6) and (5.9) hold for every $v \in V_{\mathrm{M}}$ even if v is not resource consuming. Now we can decompose R_n^b into the reward gained by the first step and the remainder. Let ς_v^t denote step $\check{\gamma} \xrightarrow{t}$, then it holds that

$$R_{n+1}^b(\varsigma_v^t\circ\pi) = \begin{cases} \rho_{\mathrm{t}}(\check{\gamma})t + \rho_{\mathrm{i}}(\check{\gamma}) + R_n^{b-\varrho(\check{\gamma})t}(\pi) & t \leq \frac{b}{\varrho(\check{\gamma})} \\ \rho_{\mathrm{t}}(\check{\gamma}) \cdot \frac{b}{\varrho(\check{\gamma})} + \rho_{\mathrm{f}}(\check{\gamma}) & t > \frac{b}{\varrho(\check{\gamma})} \end{cases} \tag{5.10}$$

We put Eq. (5.10) into (5.9). With a bit of rearranging we obtain

$$
\begin{aligned}
\mathbf{R}_{n+1}(\nu,\sigma|_{\check{\gamma}},b) = {} & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\big(\rho_{\mathrm{t}}(\check{\gamma})t+\rho_{\mathrm{i}}(\check{\gamma})\big)\int_{\pi\in\mathbb{P}}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\,\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
& + \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\int_{\pi\in\mathbb{P}} R_n^{b-\varrho(\check{\gamma})t}(\pi)\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\,\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
& + \int_{b/\varrho(\check{\gamma})}^{\infty} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\bigg(\frac{\rho_{\mathrm{t}}(\check{\gamma})b}{\varrho(\check{\gamma})}+\rho_{\mathrm{f}}(\check{\gamma})\bigg)\int_{\pi\in\mathbb{P}}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\,\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
= {} & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\big(\rho_{\mathrm{t}}(\check{\gamma})t+\rho_{\mathrm{i}}(\check{\gamma})\big)\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\int_{\pi\in\mathbb{P}}\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
& + \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\int_{\pi\in\mathbb{P}} R_n^{b-\varrho(\check{\gamma})t}(\pi)\,\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
& + \int_{b/\varrho(\check{\gamma})}^{\infty} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\bigg(\frac{\rho_{\mathrm{t}}(\check{\gamma})b}{\varrho(\check{\gamma})}+\rho_{\mathrm{f}}(\check{\gamma})\bigg)\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\int_{\pi\in\mathbb{P}}\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
= {} & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\int_{\pi\in\mathbb{P}} R_n^{b-\varrho(\check{\gamma})t}(\pi)\,\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
& + \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\big(\rho_{\mathrm{t}}(\check{\gamma})t+\rho_{\mathrm{i}}(\check{\gamma})\big)\,\mathrm{d}t + \int_{b/\varrho(\check{\gamma})}^{\infty} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\bigg(\frac{\rho_{\mathrm{t}}(\check{\gamma})b}{\varrho(\check{\gamma})}+\rho_{\mathrm{f}}(\check{\gamma})\bigg)\mathrm{d}t \\
& \hfill \Big[\ast\ \text{By } \textstyle\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu') = \int_{\pi\in\mathbb{P}}\mathfrak{p}_{\nu',\sigma[\varsigma_\nu^t]}(\mathrm{d}\pi) = 1\ \ast\Big] \\
= {} & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\mathbb{E}_{\nu',\sigma[\varsigma_\nu^t]}(R_n^{b-\varrho(\check{\gamma})t})\,\mathrm{d}t \\
& + \bigg(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\bigg)\bigg(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\bigg)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \qquad (5.11)
\end{aligned}
$$

Putting Eq. (5.11) into (5.6) with some simplification gives

$$
\overline{\underline{\mathbf{R}}}_{n+1}(\nu,b) = \operatorname*{opt}_{\sigma\in\Sigma}\int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{\nu'\in V}\mathrm{d}_{\check{\gamma}}(\nu')\,\mathbb{E}_{\nu',\sigma[\varsigma_\nu^t]}(R_n^{b-\varrho(\check{\gamma})t})\,\mathrm{d}t
$$

$$
\begin{aligned}
&+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\\
\overset{(\dagger)}{=}\;&\int_0^{{}^{b}/_{\varrho(\check{\gamma})}}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\operatorname*{opt}_{\sigma\in\Sigma}\mathbb{E}_{v',\sigma[\varsigma_v^t]}(R_n^{b-\varrho(\check{\gamma})t})\,\mathrm{d}t\\
&+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\\
\overset{(\ddagger)}{=}\;&\int_0^{{}^{b}/_{\varrho(\check{\gamma})}}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}_n\big(v',b-\varrho(\check{\gamma})t\big)\,\mathrm{d}t\\
&+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \qquad (5.12)
\end{aligned}
$$

which shows the correctness of Eq. (5.3a). Since the derivations of (†) and (‡) are nontrivial, they are explained in more details. The former is the result of interchanging of opt, the integral and sum. They can be swapped, intuitively as the integral is over t and the sum is over v', nevertheless $\sigma[\varsigma_v^t]$ can be seen as a function in both variables. More precisely, it is possible to combine the optimal strategies for each choice of t and v' into a strategy that is optimal for all possible values of t and v'. To put it formally, fix a time point $t\in[0,{}^{b}/_{\varrho(\gamma)}]$ and a successor of v, say $v'\in V$, then let $\{\sigma_k^{t,v'}\}_{k\in\mathbb{N}}$ be a sequence of strategies such that for all $k\in\mathbb{N}$

$$
\mathbb{E}_{v',\sigma_k^{t,v'}[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)\le\mathbb{E}_{v',\sigma_{k+1}^{t,v'}[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)
$$

when opt = sup. The above order must be reversed (decreasing) for the infimum case. Moreover,

$$
\lim_{k\to\infty}\mathbb{E}_{v',\sigma_k^{t,v'}[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)=\operatorname*{opt}_{\sigma\in\Sigma}\mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big) \qquad (5.13)
$$

Note that such a sequence must exist as the set of strategies is nonempty. Now we combine all the strategies indexed by t and v' to construct a new sequence, that is $\{\sigma_k\}_{k\in\mathbb{N}}$ with $\sigma_k(\gamma\xrightarrow{t}\pi)=\sigma_k^{t,\mathrm{v}_0^\pi}(\pi)$ for $t\in[0,\frac{b}{\varrho(\gamma)}]$, $k\in\mathbb{N}$ and $\pi\in\mathbb{P}$. Therefore it holds by construction that

$$
\begin{aligned}
&\int_0^{{}^{b}/_{\varrho(\check{\gamma})}}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\operatorname*{opt}_{\sigma\in\Sigma}\mathbb{E}_{v',\sigma[\varsigma_v^t]}(R_n^{b-\varrho(\check{\gamma})t})\,\mathrm{d}t\\
&\quad=\int_0^{{}^{b}/_{\varrho(\check{\gamma})}}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\lim_{k\to\infty}\mathbb{E}_{v',\sigma_k^{t,v'}[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)\,\mathrm{d}t\quad\big[*\text{ Eq. (5.13) }*\big]
\end{aligned}
$$

$$= \lim_{k\to\infty} \int\limits_0^{{}^b/_{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v'\in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathbb{E}_{v',\sigma_k^{t,v'}[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)\,\mathrm{d}t \;\big[* \text{ Thm. 2.19 } *\big]$$

$$\overset{(*)}{=} \lim_{k\to\infty} \int\limits_0^{{}^b/_{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v'\in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathbb{E}_{v',\sigma_k[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)\,\mathrm{d}t \tag{5.14}$$

where $(*)$ follows from the sequence construction. On the other hand, it holds in general that the LHS of (†) is less or equal to the RHS of (†) when $\mathrm{opt} = \sup$[1]. This fact together with Eq. (5.14) approves that the inequality must be tight.

To justify (‡), it is required to show that

$$\overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})t\big) = \operatorname*{opt}_{\sigma\in\Sigma} \mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)$$

Take any sequence of strategies $\{\sigma_k : k \in \mathbb{N}\}$ such that

$$\lim_{k\to\infty} \mathbb{E}_{v',\sigma_k}\big(R_n^{b-\varrho(\check{\gamma})t}\big) = \overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})t\big)$$

Put sequence $\{\sigma'_k : k \in \mathbb{N}\}$ with $\sigma'_k(\check{\gamma} \xrightarrow{t} \pi) := \sigma_k(\pi)$ for every $k \in \mathbb{N}$ and $\pi \in \mathbb{P}$. This construction gives

$$\overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})t\big) = \lim_{k\to\infty} \mathbb{E}_{v',\sigma_k}\big(R_n^{b-\varrho(\check{\gamma})t}\big) = \lim_{k\to\infty} \mathbb{E}_{v',\sigma'_k[\varsigma_v^t]}\big(R_n^{b-\varrho(\check{\gamma})t}\big)$$

which shows the correctness of (‡). It completes the prove of Eq. (5.3a).

Similarly to resource consuming states, for $v \in V_{\mathrm{rc}} \cup V_{\mathrm{M}}$ there is only one outgoing transition, say $\check{\gamma}$, which is indeed Markovian. Therefore, it satisfies Eq. (5.9). Decomposing R_n^b into the rewards gained by the first step and others gives

$$R_{n+1}^b(\varsigma_v^t \circ \pi) = \rho_{\mathrm{t}}(\check{\gamma})t + \rho_{\mathrm{i}}(\check{\gamma}) + R_n^b(\pi) \tag{5.15}$$

where $\varsigma_v^t := \check{\gamma} \xrightarrow{t}$. Put Eq. (5.15) into (5.9) with a bit of rearranging to obtain

$$\begin{aligned}
\mathbf{R}_{n+1}(v,\sigma|_{\check{\gamma}}, b) &= \int\limits_0^{\infty} \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t}\big(\rho_{\mathrm{t}}(\check{\gamma})t + \rho_{\mathrm{i}}(\check{\gamma})\big) \int_{\pi\in\mathbb{P}} \sum_{v'\in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathfrak{p}_{v',\sigma[\varsigma_v^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
&\quad + \int\limits_0^{\infty} \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \int_{\pi\in\mathbb{P}} R_n^b(\pi) \sum_{v'\in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathfrak{p}_{v',\sigma[\varsigma_v^t]}(\mathrm{d}\pi)\,\mathrm{d}t \\
&= \int\limits_0^{\infty} \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t}\big(\rho_{\mathrm{t}}(\check{\gamma})t + \rho_{\mathrm{i}}(\check{\gamma})\big) \sum_{v'\in V} \mathsf{d}_{\check{\gamma}}(v') \int_{\pi\in\mathbb{P}} \mathfrak{p}_{v',\sigma[\varsigma_v^t]}(\mathrm{d}\pi)\,\mathrm{d}t
\end{aligned}$$

[1] The reverse holds for the infimum case.

$$
\begin{aligned}
&+ \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \int_{\pi \in \mathbb{P}} R_n^b(\pi)\, \mathfrak{p}_{v',\sigma[\varsigma_v^t]}(\mathrm{d}\pi)\, \mathrm{d}t \\
&= \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \big(\rho_{\mathrm{t}}(\check{\gamma}) t + \rho_{\mathrm{i}}(\check{\gamma})\big)\, \mathrm{d}t + \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^b\big)\, \mathrm{d}t \\
&\hfill \Big[\ast\ \textstyle\sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') = \int_{\pi \in \mathbb{P}} \mathfrak{p}_{v',\sigma[\varsigma_v^t]}(\mathrm{d}\pi) = 1\ \ast\Big] \\
&= \frac{\rho_{\mathrm{t}}(\gamma)}{\mathsf{r}_\gamma} + \rho_{\mathrm{i}}(\gamma) + \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^b\big)\, \mathrm{d}t
\end{aligned}
\tag{5.16}
$$

Putting Eq. (5.16) into (5.6) with some simplification gives

$$
\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}(v,b) &= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \operatorname*{opt}_{\sigma \in \Sigma} \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^b\big)\, \mathrm{d}t \\
&\overset{(\dagger)}{=} \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \operatorname*{opt}_{\sigma \in \Sigma} \mathbb{E}_{v',\sigma[\varsigma_v^t]}\big(R_n^b\big)\, \mathrm{d}t \\
&\overset{(\ddagger)}{=} \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v',b\big)\, \mathrm{d}t \\
&= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v',b\big) \int_0^\infty \mathsf{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} t}\, \mathrm{d}t \\
&= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v',b\big)
\end{aligned}
$$

which confirms Eq. (5.3b). Note that (†) and (‡) can be justified with similar arguments given for the corresponding cases during the derivation of Eq. (5.12) on p. 81.

It remains to derive Eq.(5.3c) for $v \in V_{\mathrm{P}}$. In contrast to the previous cases, here v may have more that one outgoing transition. However, the one who is selected by the strategy must be executed immediately. We first compute $\mathbf{R}_{n+1}(v, \sigma|_\gamma, b)$ for some $\sigma \in \Sigma$ and $\gamma \in T(v)$. For that we use Thm. 3.31 to split the expectation at the first executed step. Once more we stress on the facts that γ is probabilistic and thereby urgent and $\sigma|_\gamma$ selects γ from v with probability one. Therefore, $\varsigma_\gamma := (\gamma, 0)$ is the only probable step that can be executed at the beginning from v under strategy $\sigma|_\gamma$, namely $\mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\{\varsigma_\gamma\}) = 1$. We can further decompose $R^b_{n+1}(\varsigma_\gamma \circ \pi)$ into the rewards gained by ς_γ and π.

$$
R^b_{n+1}(\varsigma_\gamma \circ \pi) = \rho_{\mathrm{i}}(\gamma) + R^b_n(\pi) \tag{5.17}
$$

Moreover, it holds by Defs. 3.13 and 3.15 that $\sigma[\varsigma_\gamma]$ and $\sigma|_\gamma[\varsigma_\gamma]$ are the same. Therefore,

$$
\begin{aligned}
\mathbf{R}_{n+1}(v,\sigma|_\gamma,b) &= \int_{\varsigma\in\mathbb{S}}\int_{\pi\in\mathbb{P}} R^b_{n+1}(\varsigma\circ\pi)\sum_{v'\in V}\mathsf{d}^{\varsigma}_0(v')\,\mathfrak{p}_{v',\sigma|_\gamma[\varsigma]}(\mathrm{d}\pi)\,\mathfrak{p}^{\mathbb{S}}_{v,\sigma|_\gamma}(\mathrm{d}\varsigma) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \big[*\ \text{Thm. 3.31}\ *\big] \\
&= \int_{\pi\in\mathbb{P}} R^b_{n+1}(\varsigma_\gamma\circ\pi)\sum_{v'\in V}\mathsf{d}^{\varsigma_\gamma}_0(v')\,\mathfrak{p}_{v',\sigma|_\gamma[\varsigma_\gamma]}(\mathrm{d}\pi) \\
&= \int_{\pi\in\mathbb{P}} R^b_{n+1}(\varsigma_\gamma\circ\pi)\sum_{v'\in V}\mathsf{d}_\gamma(v')\,\mathfrak{p}_{v',\sigma[\varsigma_\gamma]}(\mathrm{d}\pi) \\
&= \int_{\pi\in\mathbb{P}} \big(\rho_{\mathrm{i}}(\gamma)+R^b_n(\pi)\big)\sum_{v'\in V}\mathsf{d}_\gamma(v')\,\mathfrak{p}_{v',\sigma[\varsigma_\gamma]}(\mathrm{d}\pi) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \big[*\ \text{Eq. (5.17)}\ *\big] \\
&= \sum_{v'\in V}\mathsf{d}_\gamma(v')\int_{\pi\in\mathbb{P}} \big(\rho_{\mathrm{i}}(\gamma)+R^b_n(\pi)\big)\,\mathfrak{p}_{v',\sigma[\varsigma_\gamma]}(\mathrm{d}\pi) \\
&= \rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_\gamma(v')\int_{\pi\in\mathbb{P}} R^b_n(\pi)\,\mathfrak{p}_{v',\sigma[\varsigma_\gamma]}(\mathrm{d}\pi) \\
&\qquad\qquad\qquad \big[*\ \textstyle\sum_{v'\in V}\mathsf{d}_\gamma(v')=\int_{\pi\in\mathbb{P}}\mathfrak{p}_{v',\sigma[\varsigma_\gamma]}(\mathrm{d}\pi)=1\ *\big] \\
&= \rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_\gamma(v')\,\mathbb{E}_{v',\sigma[\varsigma_\gamma]}\big(R^b_n\big) \qquad (5.18)
\end{aligned}
$$

Putting Eq. (5.18) into (5.4) gives

$$
\begin{aligned}
\overline{\mathbf{R}}_{n+1}(v,b) &= \max_{\gamma\in T(v)}\sup_{\sigma\in\Sigma}\mathbf{R}_{n+1}(v,\sigma|_\gamma,b) && \big[*\ \text{Eq. (5.4)}\ *\big] \\
&= \max_{\gamma\in T(v)}\sup_{\sigma\in\Sigma}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_\gamma(v')\,\mathbb{E}_{v',\sigma[\varsigma_\gamma]}\big(R^b_n\big)\Big) && \big[*\ \text{Eq. (5.18)}\ *\big] \\
&\overset{(\dagger)}{=} \max_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_\gamma(v')\sup_{\sigma\in\Sigma}\mathbb{E}_{v',\sigma[\varsigma_\gamma]}\big(R^b_n\big)\Big) \\
&\overset{(\ddagger)}{=} \max_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_\gamma(v')\,\overline{\mathbf{R}}_n(v',b)\Big) && (5.19)
\end{aligned}
$$

where (†) and (‡) can be justified with similar but this time simpler arguments than the ones given for the corresponding cases during the derivation of Eq. (5.12) on p. 81. We can in the same way derive Eq. (5.19) for the infimum case. It confirms the correctness of Eq. (5.3c) and subsequently completes the proof. □

Lem. 5.1 establishes a recursive integral equation system for sequence of functions $\{\overline{\underline{\mathbf{R}}}_n\}_{n\in\mathbb{N}}$, each computing the optimal n-step ERR. Moreover, we have shown in Ch. 4 that the limit of the sequence exists and coincides with the optimal ERR. We exploit both facts to first show the continuity of the functions in the sequence and then derive the continuity is carried over into the limit. This will be discussed in more details in Sec. 5.2.

Another important result emerging from the aforementioned facts allows us to characterise the optimal ERR as a fixed point. The characterisation demonstrates that the recursion given in Lem. 5.1 has a least fixed point, which is indeed the optimal ERR. Moreover, it formulates the optimal ERR as the solution of a Volterra integral equation induced from it. The result is described in the following theorem.

Theorem 5.2. *Let $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ be an MRA, then, for every $v \in V$, $\overline{\underline{\mathbf{R}}}$ is the* least fixed point *of higher order operator $\overline{\underline{\Omega}} : (V \times \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}) \to (V \times \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0})$ where*

$$\overline{\underline{\Omega}}(F)(v,b) = \int_0^{b/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot F\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\,\mathrm{d}t \\ + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\Big) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \tag{5.20a}$$

when $v \in V_{\mathrm{rc}}$ with $T(v) = \{\check{\gamma}\}$,

$$\overline{\underline{\Omega}}(F)(v,b) = \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot F(v', b) \tag{5.20b}$$

if $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$, and finally for $v \in V_{\mathrm{P}}$:

$$\overline{\underline{\Omega}}(F)(v,b) = \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v') \cdot F(v', b)\Big) \tag{5.20c}$$

Proof. The proof proceeds in two steps. We first claim that the optimal ERR is a fixed point of $\overline{\underline{\Omega}}$. Afterwards, we show that it is the least fixed point. To show the former observe that $\lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_n = \overline{\underline{\mathbf{R}}}$ by Lem. 4.11 and 4.13. Moreover, by comparing Eq. (5.3a), (5.3b) and (5.3c) respectively with Eq. (5.20a), (5.20b) and (5.20c), it turns out that $\overline{\underline{\mathbf{R}}}_{n+1} = \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}_n\big)$, for all $n \in \mathbb{N}$. It is therefore enough to show for every $v \in V$ and that

$$\begin{aligned} \overline{\underline{\mathbf{R}}}(v,b) = \lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_{n+1}(v,b) = \lim_{n \to \infty} \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}_n\big)(v,b) \\ = \overline{\underline{\Omega}}\big(\lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_n\big)(v,b) \\ = \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}\big)(v,b) \end{aligned} \tag{5.21}$$

The crucial part is to justify that Eq. (5.21) holds, i. e. the limit and $\overline{\underline{\Omega}}$ operators are interchangeable. More specifically, it must be possible to interchange the limit operator with the integral, the summation and the minimum/maximum operators given in Eq. (5.20a), (5.20b) and (5.20c). To explain why the interchanges are valid we first provide general arguments. For this we heavily use the fact that the sequence $\{\underline{\overline{\mathbf{R}}}_n\}_{n\in\mathbb{N}}$ is monotonically increasing. The interchange of the limit with the integral and the summation lying in Eq. (5.20a) and (5.20b) is justified by the monotone convergence theorem. Furthermore, the limit and the minimum/maximum in Eq. (5.20c) can be interchanged since the sequence of functions inside the mix operator are uniformly convergent on T. We explain each of these arguments in more details next.

We first consider the interchange between the limit and the summation. Observe that $\underline{\overline{\mathbf{R}}}_n(v,b) \le \underline{\overline{\mathbf{R}}}_{n+1}(v,b)$ for every $v \in V$, $b \in \mathbb{R}_{\ge 0}$ and $n \in \mathbb{N}$. Moreover, summation can be viewed as integration with respect to counting measures. Hence, the monotone convergence theorem (Thm. 2.19) together with Lem. 4.11 and 4.13 implies that

$$\lim_{n\to\infty}\sum_{v'\in V}\mathrm{d}_{\gamma}(v')\,\underline{\overline{\mathbf{R}}}_n(v',b) = \sum_{v'\in V}\lim_{n\to\infty}\mathrm{d}_{\gamma}(v')\,\underline{\overline{\mathbf{R}}}_n(v',b) = \sum_{v'\in V}\mathrm{d}_{\gamma}(v')\,\underline{\overline{\mathbf{R}}}(v',b) \tag{5.22}$$

With a similar argument we can switch between the limit and the integration. Putting all facts together, we can write

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}(v,b) &= \lim_{n\to\infty}\underline{\overline{\mathbf{R}}}_{n+1}(v,b)\\
&= \lim_{n\to\infty}\int_0^{b/\varrho(\check{\gamma})}\mathrm{r}_{\check{\gamma}}\cdot\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}_n\big(v',b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \qquad \big[*\text{ Eq. (5.3a) }*\big]\\
&= \int_0^{b/\varrho(\check{\gamma})}\lim_{n\to\infty}\mathrm{r}_{\check{\gamma}}\cdot\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}_n\big(v',b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \qquad \big[*\text{ Thm. 2.19 }*\big]\\
&= \int_0^{b/\varrho(\check{\gamma})}\mathrm{r}_{\check{\gamma}}\cdot\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}\big(v',b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \qquad \big[*\text{ Eq. (5.22) }*\big]\\
&= \underline{\overline{\Omega}}\big(\underline{\overline{\mathbf{R}}}\big)(v,b)
\end{aligned}$$

which justifies Eq. (5.21) for $v \in V_{\mathrm{rc}}$. Similarly for $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$ we have

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}(v,b) &= \lim_{n\to\infty}\underline{\overline{\mathbf{R}}}_{n+1}(v,b)\\
&= \lim_{n\to\infty}\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})+\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}_n(v',b)\Big)\\
&= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})+\lim_{n\to\infty}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}_n(v',b)
\end{aligned}$$

$$\begin{aligned}
&= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}(v', b) && \big[* \text{ Eq. (5.22) } *\big] \\
&= \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}\big)(v, b)
\end{aligned}$$

It remains to prove the correctness of Eq. (5.21) for $v \in V_{\mathrm{P}}$. For this we make use of the fact that the set of transitions is finite and thereby by Thm. 2.22 the convergence of

$$p_n^{v,b}(\gamma) = \rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}_n(v', b)$$

is uniform on the set of transitions $T(v)$ for $v \in V_{\mathrm{P}}$. We can thus write

$$\begin{aligned}
\overline{\underline{\mathbf{R}}}(v, b) &= \lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_{n+1}(v, b) \\
&= \lim_{n \to \infty} \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}_n(v', b) \Big) \\
&= \operatorname*{mix}_{\gamma \in T(v)} \lim_{n \to \infty} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}_n(v', b) \Big) && \big[* \text{ Thm. 2.22 and 2.21 } *\big] \\
&= \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \lim_{n \to \infty} \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}_n(v', b) \Big) \\
&= \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}(v', b) \Big) && \big[* \text{ Eq. (5.22) } *\big] \\
&= \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}\big)(v, b)
\end{aligned}$$

In the second step, we show that $\overline{\underline{\mathbf{R}}}$ is the least fixed point of $\overline{\underline{\Omega}}$. Let $G : V \times \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}$ be an arbitrary fixed point of $\overline{\underline{\Omega}}$, then we prove, by induction on n, that $\overline{\underline{\mathbf{R}}}_n \leq G$ for all $n \in \mathbb{N}$. For the base case, it obviously holds that $\mathbf{0} = \overline{\underline{\mathbf{R}}}_0 \leq G$. Assume that $\overline{\underline{\mathbf{R}}}_n \leq G$ holds for some $n \geq 0$. It gives

$$\overline{\underline{\mathbf{R}}}_{n+1} \leq \overline{\underline{\Omega}}\big(\overline{\underline{\mathbf{R}}}_n\big) \overset{(\dagger)}{\leq} \overline{\underline{\Omega}}\big(G\big) = G$$

where (†) follows from the fact that $\overline{\underline{\Omega}}$ is order preserving. That is to say, if $F_1 \leq F_2$ then $\overline{\underline{\Omega}}(F_1) \leq \overline{\underline{\Omega}}(F_2)$. This can be easily shown for each case of the theorem described by Eq. (5.20a), (5.20b) and (5.20c). The final result is obtained by taking the limit of both sides of the inequality, i. e. $\overline{\underline{\mathbf{R}}} = \lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_n \leq G$. □

The characterisation offered by Thm. 5.2 provides the opportunity for better understanding of how the optimal ERR behaves. This will be used for studying the continuity and the differentiability of the optimal ERR as a function of resource budget. Furthermore, this forms the basis for computing the optimal ERR in an algorithmic way.

It is important to note that a larger resource budget in the optimal ERR does not necessarily mean a higher reward gain. The converse in particular occurs when the final reward of a resource consuming state is very large. The tighter the resource budget in this case, the more likely to reside in the state at the moment of running over the budget and thereby the larger the reward to gain. In the next section however, we study a certain part of the optimal ERR that is increasing with respect to the resource budget. This will be used for proving the Lipschitz continuity.

5.2 Continuity

In this section we study the continuity of the optimal ERR as a function of resource bound. There are different notions of continuity in mathematics, among them we study Lipschitz continuity, as described in Def. 2.17. It is a strong notion of continuity that, in addition to implying the conventional form of continuity, it limits how fast a function can vary. Lipschitz bounds the absolute value of the slope between two arbitrary points on the function. The bound is called *Lipschitz constant*. One important property of Lipschitz functions is that they can be approximated arbitrarily closely by discretisation, since their maximum speed of variation is bounded. We use this feature to prove the error bound of the discretisation algorithm given in Ch. 6.

We show in this section that the optimal ERR of NRC models is Lipschitz continuous. This intuitively means that by varying the resource budget, the variation in the optimal ERR is bounded. In other words, the optimal ERR cannot rise or drop endlessly fast. Moreover, we establish the bound for the variation, which is identified by a Lipschitz constant. The Lipschitz constant is expressed in terms of some parameters of the model. In the sequel we first show that the optimal n-step ERRs are Lipschitz with a global constant. Such a sequence in general enjoys an interesting property, namely its limit, which is in our case the optimal ERR, is also Lipschitz continuous with the same constant. The result mainly relies upon the fact that the optimal ERR is bounded (Cor. 4.23.1) and the characterisation of the optimal n-step ERR (Lem. 5.1).

At the first step we investigate how the optimal n-step ERR changes as the resource budget decreases or increases. The next lemma studies the variation of $\overline{\underline{\mathbf{R}}}_n(v,\cdot)$ under a resource budget change, for $v \in V_{\mathrm{rc}}$. It provides a basis for establishing Lipschitz continuity of the optimal n-step ERR.

Lemma 5.3. *For $v \in V_{\mathrm{rc}}$ with $T(v) = \{\check{\gamma}\}$ and $0 \leq \delta \leq b$ it holds for all $n \in \mathbb{N}$ that:*

$$\begin{aligned}\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = & \int_{0}^{\delta/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\,\mathrm{d}t \\ & + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot \delta}{\varrho(\check{\gamma})}}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}(v, b - \delta)\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot \delta}{\varrho(\check{\gamma})}} \qquad (5.23)\end{aligned}$$

Proof. Eq. (5.23) can be derived via Eq. (5.3a) in Lem. 5.1. Let $F_n(\check{\gamma}, b)$ be the integral given in Eq. (5.3a). We split the integral by dividing the interval $[0, \frac{b}{\varrho(\gamma)}]$ at point $\frac{\delta}{\varrho(\gamma)}$.

$$\begin{aligned}
F_n(\check{\gamma}, b) &= \int_0^{b/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&= \int_0^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&\quad + \int_{\frac{\delta}{\varrho(\check{\gamma})}}^{b/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&\overset{(*)}{=} \int_0^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&\quad + \int_0^{\frac{b-\delta}{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot (\tau + \delta)} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \delta - \varrho(\check{\gamma}) \cdot \tau\big)\, \mathrm{d}\tau \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Big[*\ \tau = t - \tfrac{\delta}{\varrho(\check{\gamma})}\ *\Big] \\
&= \int_0^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&\quad + \mathsf{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot \delta}{\varrho(\check{\gamma})}} \int_0^{\frac{b-\delta}{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot \tau} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \delta - \varrho(\check{\gamma})\tau\big)\, \mathrm{d}\tau \qquad (5.24)
\end{aligned}$$

where $(*)$ is derived by substituting t in the second integral by $\tau = t - \frac{\delta}{\varrho(\gamma)}$. Put Eq. (5.24) into Eq. (5.3a) to obtain:

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}_{n+1}(v, b) &= \int_0^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\
&\quad + \mathsf{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot \delta}{\varrho(\check{\gamma})}} \int_0^{\frac{b-\delta}{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}} \cdot \mathsf{e}^{-\mathsf{r}_{\check{\gamma}} \cdot \tau} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v') \cdot \underline{\overline{\mathbf{R}}}_n\big(v', b - \delta - \varrho(\check{\gamma})\tau\big)\, \mathrm{d}\tau \\
&\quad + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathsf{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\Big) + \rho_{\mathrm{f}}(\check{\gamma})\, \mathsf{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}
\end{aligned}$$

$$
\begin{aligned}
&= \int_{0}^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
&\quad + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} \int_{0}^{\frac{b-\delta}{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot\tau} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b-\delta-\varrho(\check{\gamma})\tau\big)\,\mathrm{d}\tau \\
&\quad + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} - \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) \\
&\quad + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} \cdot \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}(b-\delta)}{\varrho(\check{\gamma})}} \\
&= \int_{0}^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
&\quad + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}}\Big) \\
&\quad + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} \int_{0}^{\frac{b-\delta}{\varrho(\check{\gamma})}} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot\tau} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b-\delta-\varrho(\check{\gamma})\tau\big)\,\mathrm{d}\tau \\
&\quad + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}}\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}(b-\delta)}{\varrho(\check{\gamma})}}\Big) \\
&\quad + \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}} \cdot \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}(b-\delta)}{\varrho(\check{\gamma})}} \\
&\overset{(\dagger)}{=} \int_{0}^{\delta/\varrho(\check{\gamma})} \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v') \cdot \overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
&\quad + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}(v, b-\delta)\,\mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}}\cdot\delta}{\varrho(\check{\gamma})}}
\end{aligned}
$$

$\big[* \text{ Eq. (5.3a) } *\big]$

□

Lem. 5.3 helps us to inspect the variation of the optimal n-step ERR under resource budget changes. In order to establish their Lipschitz continuity we need to utilise an auxiliary inequality. The inequality demonstrate an upper bound for the optimal n-step ERR with respect to the optimal *expected n-step resource-bounded cumulative reward*, denoted by $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_n$. Intuitively speaking, $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_n$ computes the optimal n-step ERR without considering the final rewards.

Definition 5.4. *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_n$ is the optimal n-step ERR ($\overline{\underline{\mathbf{R}}}_n$) of $\mathcal{R}$ after replacing ρ_{f} with zero function, i. e. $\rho_{\mathrm{f}} := \mathbf{0}$.*

As the definition suggests $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}$ is a special case of the optimal ERR. Hence, it inherits the properties of the optimal ERR. In particular, the results of Lem. 5.1 and 5.3 holds also for $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}$. Moreover, one can view $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}$ as partial rewards gained by transient and instantaneous but not final rewards. In this regard we utilise similar technique as used in Sec. 4.3.1 to decompose the optimal n-step ERR into $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}$ and $\overline{\underline{\mathbf{R}}}_n^{\mathrm{f}}$. The latter, the optimal *expected n-step resource-bounded final reward*, is always less than or equal to $\bar{\rho}_{\mathrm{f}}$. The next lemma associates $\overline{\underline{\mathbf{R}}}_n$ with $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}$ for a resource consuming state via an inequality.

Lemma 5.5. *For a given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ and $b \in \mathbb{R}_{\geq 0}$ it holds for every with $v \in V_{\mathrm{rc}}$ with $T(v) = \{\check{\gamma}\}$ and $n \in \mathbb{N}$ that*

$$\overline{\underline{\mathbf{R}}}_n(v,b) \leq \overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,b) + \bar{\rho}_{\mathrm{f}} \cdot \left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}$$

Proof. As the optimal n-step ERR consists of cumulative and final rewards it is not hard to see that $\overline{\underline{\mathbf{R}}}_n(v,b) \leq \overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,b) + \bar{\rho}_{\mathrm{f}}$. Then, observe that the result holds for $n = 0$ since $\overline{\underline{\mathbf{R}}}_0$ and $\overline{\underline{\mathbf{R}}}_0^{\mathrm{c}}$ are constantly zero everywhere. We then turn our attention to $n+1$ step objectives for $n \in \mathbb{N}$. From Eq. (5.3a) we have

$$\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\,\mathrm{d}t \\
& + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
\leq & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v') \left(\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v', b - \varrho(\check{\gamma}) \cdot t) + \bar{\rho}_{\mathrm{f}}\right)\mathrm{d}t \\
& + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
= & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\,\mathrm{d}t \\
& + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \bar{\rho}_{\mathrm{f}} \cdot \left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
= & \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c}}(v,b) + \bar{\rho}_{\mathrm{f}} \cdot \left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \qquad \big[* \text{ Eq. (5.3a) } *\big]
\end{aligned}$$

□

Now we are ready to establish the Lipschitz continuity of the optimal n-step ERR. The next lemma shows that $\overline{\underline{\mathbf{R}}}_n$'s are uniformly Lipschitz with a global constant, meaning that the speed of their variation is globally bounded for all $n \in \mathbb{N}$. The result is indeed limited to NRC models, as the optimal ERR of an NRD MRA might reach infinity.

Lemma 5.6. *Given NRC MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, then $\overline{\underline{\mathbf{R}}}_n(v,\cdot)\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is Lipschitz continuous with constant $L = \frac{1}{\underline{\varrho}}\big(\bar{\rho}_\mathrm{t} + \bar{\lambda}(\bar{\rho}_\mathrm{i} + r_0 + \bar{\rho}_\mathrm{f})\big)$ for all $v \in V$ and $n \in \mathbb{N}$.*

Proof. We proceed by induction on n and show for every $0 \leq b \leq b'$ that

$$-\bar{\rho}_\mathrm{f}\frac{\bar{\lambda}}{\underline{\varrho}}(b'-b) \leq \overline{\underline{\mathbf{R}}}_n(v,b') - \overline{\underline{\mathbf{R}}}_n(v,b) \leq L\cdot(b'-b) \tag{5.25}$$

It is straightforward to establish the induction base, as $\overline{\underline{\mathbf{R}}}_0(v,\cdot)$ is constant and thereby Lipschitz. We assume now that Eq. (5.25) holds for $n \in \mathbb{N}$ and then prove it for $n+1$. We distinguish between different cases given in Lem. 5.1. For $v \in V_\mathrm{rc}$ with $T(v) = \{\check{\gamma}\}$ we apply the result of Lem. 5.3 to have

$$\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}(v,b') = & \int\limits_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n\big(v', b'-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
& + \Big(\frac{\rho_\mathrm{t}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_\mathrm{i}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \\
& \hfill \big[*\text{ Eq. (5.23) }*\big] \\
\leq & \int\limits_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\big(\overline{\underline{\mathbf{R}}}_n(v', b-\varrho(\check{\gamma})\cdot t) + L\cdot(b'-b)\big)\,\mathrm{d}t \\
& + \Big(\frac{\rho_\mathrm{t}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_\mathrm{i}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \\
& \hfill \big[*\text{ by I. H. }*\big] \\
= & \int\limits_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
& + \Big(\frac{\rho_\mathrm{t}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_\mathrm{i}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) \\
& + \overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \\
= & \int\limits_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}}\cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t} \sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n\big(v', b-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t \\
& + \Big(\frac{\rho_\mathrm{t}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_\mathrm{i}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + \rho_\mathrm{f}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} - \rho_\mathrm{f}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \\
& + L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} \\
= & \,\overline{\underline{\mathbf{R}}}_{n+1}(v,b) - \rho_\mathrm{f}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}} + L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)
\end{aligned}$$

$$
\begin{aligned}
&+\overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\\
&\leq \overline{\underline{\mathbf{R}}}_{n+1}(v,b)-\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}+L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)\\
&\quad+\Big(\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,b'-b)+\bar{\rho}_{\mathrm{f}}\big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\Big)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\\
&\hfill\big[*\text{ Lem. 5.5 }*\big]
\end{aligned}
$$

Now we apply Cor. 4.23.1 and make use of the monotonicity of $\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n}$ to obtain

$$
\begin{aligned}
\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,b'-b)\leq\overline{\underline{\mathbf{R}}}^{\mathrm{c}}(v,b'-b)&\leq\frac{b'-b}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{t}}+\bar{\lambda}(\bar{\rho}_{\mathrm{i}}+r_0)\big)\\
&=(L-\bar{\rho}_{\mathrm{f}}\tfrac{\bar{\lambda}}{\underline{\varrho}})(b'-b) \qquad (5.26)
\end{aligned}
$$

We consider Eq. (5.26) with the fact that $1-\mathrm{e}^{x}\leq x$ for $x\geq 0$ and continue the above derivation

$$
\begin{aligned}
&\overline{\underline{\mathbf{R}}}_{n+1}(v,b')-\overline{\underline{\mathbf{R}}}_{n+1}(v,b)\\
&\qquad\leq -\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}+L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)\\
&\qquad\quad+\Big((L-\bar{\rho}_{\mathrm{f}}\tfrac{\bar{\lambda}}{\underline{\varrho}})(b'-b)+\bar{\rho}_{\mathrm{f}}\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\Big)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\\
&\qquad\leq -\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}+L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)\\
&\qquad\quad+\Big((L-\bar{\rho}_{\mathrm{f}}\tfrac{\bar{\lambda}}{\underline{\varrho}})(b'-b)+\bar{\rho}_{\mathrm{f}}\tfrac{\bar{\lambda}}{\underline{\varrho}}(b'-b)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\Big)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\\
&\qquad= -\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}+L\cdot(b'-b)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+L\cdot(b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\\
&\qquad\quad+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\\
&\qquad= L\cdot(b'-b)-\rho_{\mathrm{f}}(\check{\gamma})\big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b'-b)}{\varrho(\check{\gamma})}}\big)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}}\leq L\cdot(b'-b)
\end{aligned}
$$

This completes the proof of the upper bound for $v\in V_{\mathrm{rc}}$. For the lower bound, first observe that Eq. (5.3a) gives

$$
\mathbf{R}_{n+1}(v,b)\geq\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}b}{\varrho(\check{\gamma})}} \qquad (5.27)
$$

for $n\in\mathbb{N}$. We then apply Lem. (5.1) to get

$$
\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}(v,b')=&\int_0^{b/\varrho(\check{\gamma})}\mathrm{r}_{\check{\gamma}}\cdot\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\cdot t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\cdot\overline{\underline{\mathbf{R}}}_{n}\big(v',b'-\varrho(\check{\gamma})\cdot t\big)\,\mathrm{d}t\\
&+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\Big)+\overline{\underline{\mathbf{R}}}_{n+1}(v,b'-b)\,\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}\cdot b}{\varrho(\check{\gamma})}}\\
&\hfill\big[*\text{ Eq. (5.23) }*\big]
\end{aligned}
$$

$$
\begin{aligned}
&\geq \int_{0}^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v') \left(\underline{\overline{\mathbf{R}}}_n(v', b - \varrho(\check{\gamma}) \cdot t) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\right) \mathrm{d}t \\
&\quad + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \underline{\overline{\mathbf{R}}}_{n+1}(v, b' - b)\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
&\hspace{20em} \left[* \text{ by I. H. } *\right] \\
&= \int_{0}^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\overline{\mathbf{R}}}_n\left(v', b - \varrho(\check{\gamma}) \cdot t)\right) \mathrm{d}t \\
&\quad + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) \\
&\quad + \underline{\overline{\mathbf{R}}}_{n+1}(v, b' - b)\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
&= \int_{0}^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} \cdot t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\overline{\mathbf{R}}}_n\left(v', b - \varrho(\check{\gamma}) \cdot t)\right) \mathrm{d}t \\
&\quad + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} - \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
&\quad - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) + \underline{\overline{\mathbf{R}}}_{n+1}(v, b' - b)\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
&= \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) \\
&\quad + \underline{\overline{\mathbf{R}}}_{n+1}(v, b' - b)\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \hspace{10em} \left[* \text{ Eq. (5.3a) } *\right] \\
&\geq \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) \\
&\quad + \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b' - b)}{\varrho(\check{\gamma})}}\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} b}{\varrho(\check{\gamma})}} \hspace{10em} \left[* \text{ Eq. (5.27) } *\right] \\
&= \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) - \rho_{\mathrm{f}}(\check{\gamma})\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}}(b' - b)}{\varrho(\check{\gamma})}}\right)\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} b}{\varrho(\check{\gamma})}} \\
&\overset{(\dagger)}{\geq} \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) - \rho_{\mathrm{f}}(\check{\gamma}) \frac{\mathrm{r}_{\check{\gamma}}(b' - b)}{\varrho(\check{\gamma})} \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} b}{\varrho(\check{\gamma})}} \\
&\geq \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\left(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\right) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)\mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} b}{\varrho(\check{\gamma})}} \\
&= \underline{\overline{\mathbf{R}}}_{n+1}(v, b) - \bar{\rho}_{\mathrm{f}} \frac{\bar{\lambda}}{\underline{\varrho}}(b' - b)
\end{aligned}
$$

where (†) follows from $1 - \mathrm{e}^{-x} \leq x$ for $x \geq 0$. The second case corresponds to $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$. For that we employ Eq.(5.3b) to get:

$$
\underline{\overline{\mathbf{R}}}_{n+1}(v, b') - \underline{\overline{\mathbf{R}}}_{n+1}(v, b) = \sum_{v' \in V} \mathrm{d}_{\gamma}(v') \left(\underline{\overline{\mathbf{R}}}_n(v, b') - \underline{\overline{\mathbf{R}}}_n(v, b)\right)
$$

$$\begin{aligned}
&\leq \sum_{v'\in V} \mathsf{d}_{\gamma}(v')\cdot L\cdot(b'-b) && \left[*\text{ I. H. }*\right]\\
&= L\cdot(b'-b)
\end{aligned}$$

The proof of the lower bound is similar. For the last case, $v \in V_{\mathrm{P}}$, we first look into the upper bound and define

$$\begin{aligned}
\underline{\gamma} &:= \operatorname*{arg\,min}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V}\mathsf{d}_{\gamma}(v')\underline{\mathbf{R}}_n(v',b)\Big)\\
\bar{\gamma} &:= \operatorname*{arg\,max}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V}\mathsf{d}_{\gamma}(v')\overline{\mathbf{R}}_n(v',b')\Big)
\end{aligned}$$

Using Eq. (5.3c) we have

$$\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}(v,b')-\overline{\underline{\mathbf{R}}}_{n+1}(v,b) = &\operatorname*{mix}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b')\Big)\\
&-\operatorname*{mix}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b)\Big)\\
\leq\ &\rho_{\mathrm{i}}(\bar{\underline{\gamma}})+\sum_{v'\in V}\mathsf{d}_{\bar{\underline{\gamma}}}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b')\\
&-\rho_{\mathrm{i}}(\bar{\underline{\gamma}})+\sum_{v'\in V}\mathsf{d}_{\bar{\underline{\gamma}}}(v')\,\overline{\underline{\mathbf{R}}}_n(v',b)\\
=\ &\sum_{v'\in V}\mathsf{d}_{\bar{\underline{\gamma}}}(v')\,\Big(\overline{\underline{\mathbf{R}}}_n(v',b')-\overline{\underline{\mathbf{R}}}_n(v',b)\Big)\\
\leq\ &\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\cdot L\cdot(b'-b) \qquad\qquad \left[*\text{ I. H. }*\right]\\
=\ &L\cdot(b'-b)
\end{aligned}$$

The proof of the lower bound is analogous. Hence, we finally showed the correctness of Eq. (5.25). Afterwards, as $L \leq \bar{\rho}_{\mathrm{f}}\frac{\bar{\lambda}}{\underline{\varrho}}$, we can write

$$\left|\overline{\underline{\mathbf{R}}}_{n+1}(v,b')-\overline{\underline{\mathbf{R}}}_{n+1}(v,b)\right| \leq L\left|b'-b\right|$$

which completes the proof. □

Lipschitz continuity of a sequence of functions is in general closed under their pointwise limit, provided that the functions have all a global Lipschitz constant. This is indeed the case for the optimal ERR. Therefore, the Lipschitz continuity of $\overline{\underline{\mathbf{R}}}_n$ is carried over into $\overline{\underline{\mathbf{R}}}$.

Theorem 5.7. *Given NRC MRA $\mathcal{R} := (\mathcal{M}, \underline{\varrho}, \rho)$, then $\overline{\underline{\mathbf{R}}}(v,\cdot)\colon \mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$ is Lipschitz continuous with constant $L = \frac{1}{\underline{\varrho}}\big(\bar{\rho}_{\mathrm{t}}+\bar{\lambda}(\bar{\rho}_{\mathrm{i}}+r_0+\bar{\rho}_{\mathrm{f}})\big)$ for all $v\in V$.*

Proof. From Lem. 5.6 it holds that

$$\left|\overline{\underline{\mathbf{R}}}_n(v,b')-\overline{\underline{\mathbf{R}}}_n(v,b)\right| \leq L\left|b'-b\right|$$

for every $v \in V$, $n \in \mathbb{N}$ and $b', b \in \mathbb{R}_{\geq 0}$. Since the model is NRC and by Lem. 4.11 and 4.13 it holds that $\lim_{n \to \infty} \overline{\underline{\mathbf{R}}}_n(v, b) = \overline{\underline{\mathbf{R}}}(v, b) < \infty$ for all $v \in V$ and $b \in \mathbb{R}_{\geq 0}$. The claim then follows by taking the limit of both sides of the inequality when n goes to infinity. □

Being Lipschitz continuous is an important property of a function. Many other nice features comes after Lipschitz continuity. In particular, Lipschitz continuous functions can be effectively discretised since their variation is bounded. We utilise this feature for the computation of the optimal ERR in Ch. 6.

In the next section we look into the derivative of the optimal ERR.

5.3 Differentiability

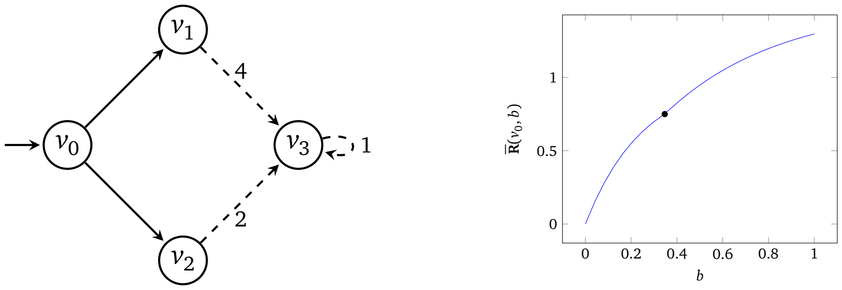
There is a connection between being Lipschitz continuous and being differentiable for real valued functions. It is known that every differentiable function with bounded first derivative is Lipschitz continuous. The converse is not always true. However, it holds that every Lipschitz continuous function is almost everywhere differentiable. It is the direct corollary of Thm. 5.7.

Corollary 5.7.1. *For an NRC MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, $\overline{\underline{\mathbf{R}}}(v, \cdot) \colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is almost everywhere differentiable for all $v \in V$.*

The corollary states that the derivative may not exist in a set with Lebesgue measure zero. In fact, such a set may exist for the optimal ERR. This set corresponds to the points in time horizon at which the optimal strategy changes the transition taken from a probabilistic state. The time instances are usually referred to as *swapping points*.

The decision changes at a swapping point as the current transition will not perform optimally afterwards. This also means at the swapping point the current and the next transition perform equally, but the latter may then overtake the former. It is thus possible that the variation of the optimal ERR is different before and after the swapping point. We illustrate an example of this situation.

The MRA depicted in Fig. 5.1a contains four states and five transitions. Markovian transition are distinguished by dashed arrows. At it is visible from the structure, the distributions of all transitions are Dirac. Among transitions, $(v_1, 4, \Delta_{v_3})$ and respectively $(v_2, 2, \Delta_{v_3})$ have transient reward of 4 and 3, i. e. $\rho_{\mathrm{t}}(v_1, 2, \Delta_{v_3}) = 3$ and $\rho_{\mathrm{t}}(v_2, 3, \Delta_{v_3}) = 4$. The reward of all other transitions is zero. We aim to compute the maximal expected time-bounded reward of the initial state v_0. For this, we assign one to the resource consumption rate of all Markovian transitions. Suppose time bound $b \in \mathbb{R}_{\geq 0}$ is given. From Thm. 5.2 it



(a) The MRA (b) The maximal ERR from v_0

Figure 5.1: MRA with a non-differentiable swapping point

holds that

$$\begin{aligned}
&\overline{\mathbf{R}}(v_0,b) = \max\big\{\overline{\mathbf{R}}(v_1,b),\overline{\mathbf{R}}(v_2,b)\big\}\\
&\overline{\mathbf{R}}(v_1,b) = 1-\mathrm{e}^{-4b}\\
&\overline{\mathbf{R}}(v_2,b) = \tfrac{3}{2}\big(1-\mathrm{e}^{-2b}\big)\\
&\overline{\mathbf{R}}(v_3,b) = 0
\end{aligned}$$

The decision of the optimal strategy changes at $t = \frac{\log(2)}{2}$ where $(v_0,\natural,\Delta_{v_2})$ is favourable to $(v_0,\natural,\Delta_{v_2})$. The evolution of $\overline{\mathbf{R}}(v_0,b)$ in terms of b is illustrated in Fig. 5.1b. As it depicted in the figure, the left and the right derivatives are different at t. It can be indeed shown that the left and the right derivative at t are respectively 1 and $\frac{3}{2}$.

5.4 Discussion

This chapter investigated certain characteristics of the optimal ERR, mainly rooted in its fixed point characterisation. Here we see the major points illustrated in this chapter.

(*i*) We proposed a fixed point characterisation of the optimal ERR as a Volterra integral equation;

(*ii*) We established the Lipschitz continuity of the optimal ERR and determined its Lipschitz constant;

(*iii*) We studied the derivative of the optimal ERR and showed that the derivative may not exist in *swapping points*.

Contributions. The fixed point characterisation mentioned in Point (*i*) is the generalisation of our previous works. We started by the fixed point characterisation for time-bounded reachability [HH12; Guc+13; Guc+14a] in MAs and time-bounded rewards [Guc+14b; Bra+15] in MRAs and extended it to resource-bounded rewards in [Hat+15]. We also studied for the first time the continuity of the optimal ERR. Here we considered a strong notion of continuity, namely Lipschitz, and showed that optimal n-step ERRs and thereby the optimal ERR are Lipschitz all with a global constant. We looked into the derivative of the optimal ERR and showed that the derivative may not exist at swapping points.

Related works. The techniques that we utilised in this chapter are inspired by a series of results mostly on CTMDPs and IMCs. Characterisation of time-bounded reachability of CTMCs as the least fixed point of a Volterra integral equation was first proposed by [Bai+03]. It was then extended by [Bai+08; Fu14a; Fu14b] to resource-bounded reachability in CTMDPs, by [Neu10; ZN10] to time-bounded reachability in CTMDPs and IMCs. Rewards are not considered by any of the mentioned works. Lipschitz continuity and differentiability of resource-bounded reachability in CTMDPs are discussed by [Fu14a; Fu14b]. Furthermore, the conventional notion of continuity is studied for time-bounded reachability of CTMDPs by [Neu10].

Future works. We did not look into the differential characterisation of the optimal ERR. It is indeed possible to represent the optimal ERR as an ordinary differential equation. This view does help to investigate the uniqueness of the fixed point discussed in Thm. 5.2. We strongly believe that at least for MRAs without *timelock* the fixed point is unique and can be computed by solving the equivalent “initial value problem”. Another worthwhile extension is to inspect the existence and then the structure of the optimal strategy. Similar works have been done for resource- and time-bounded reachability in CTMDPs [Fu14b; Neu10]. Furthermore, it has been shown the optimal strategy for time-bounded reward [Mil68] and time-bounded reachability [RS11] of CTMDPs has finitely many swapping points. We believe that the optimal strategy for ERR in MRAs has the same shape.

Part III

Numerical Methods and Applications

Chapter 6

Numerical Computations

This chapter provides tools and algorithms to compute the optimal ERR for an MRA. We start with a slightly simpler problem, namely the *optimal expected time-bounded reward* (ETR). It is simply the optimal ERR under the consideration of time as the resource. We propose a stable numerical solution based on discretisation for computing an approximation of the objective, which in addition offers a strict error bound. Using the discretisation approach the objective can be approximated arbitrarily close to the exact value. Based on this approach we develop an algorithm for computing the optimal ETR. Furthermore, we study the complexity of the algorithm and show that our solution can compute the approximate optimal ETR in polynomial time.

At the end of this chapter, we address the computation of the optimal ERR. We introduce a reduction from resource- to time-bounded objectives via a measure preserving model transformation. The transformation is linear in the size of the model and thereby efficiently applicable in practice. In general, the transformation suggests that any algorithm that computes the optimal ETR can also compute the optimal ERR.

6.1 From resource to time

The *optimal expected time-bounded reward* (ETR) estimates the optimal expected reward gained until a deadline is met. It imposes a restriction on the duration of the reward collection. From another point of view, it is a special case of the optimal ERR when Markovian transitions all consume one units of resource per time unit. In this case, the resource consumption is naturally equivalent to the passage of time.

Definition 6.1 (Optimal ETR). *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$, the* optimal expected time-bounded reward*, denoted by $\overline{\underline{\mathbf{R}}}$, is the optimal ERR under resource function $\varrho := \mathbb{1}_{T_{\mathrm{M}}}$.*

Def. 6.1 provides the specific resource function for the time-bounded setting. In addition, it implies that all random variables and expectations that are defined for the optimal ERR can be carried over into the time-bounded setting by considering the resource function. We use R^b, R^b_n, $\mathbf{R}$ and $\mathbf{R}_n$ for $n \in \mathbb{N}$ to denote the respective random variables and expectations in the time-bounded setting. Note that $b \in \mathbb{R}_{\geq 0}$ in these cases is specifically a time bound rather than a general resource budget.

State and transition partitioning is simpler in the time-bounded setting. In this case, as Markovian transitions all are resource consumers with rate one, being Markovian is equivalent to being resource consuming. Therefore, we simply partition both state and transition spaces into Markovian respectively probabilistic ones, as before, and safely ignore the partitioning based on resource consumption. The latter is indeed implied by the former.

Having in mind that ETR is a special case of ERR, all measurability and results in the resource-bounded setting given in Ch. 4 and 5 continue to be valid in the time-bounded setting, i. e. when the resource function $\mathbb{1}_{T_{\mathrm{M}}}$ is set. In particular, the fixed point characterisation of the optimal ERR can be adapted for the time-bounded setting. We obtain as a particular case of Thm. 5.2 the following fixed point characterisation for the optimal ETR.

Theorem 6.2. *For MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ with $\varrho := \mathbb{1}_{T_{\mathrm{M}}}$, $\underline{\overline{\mathbf{R}}}$ is the* least fixed point *of higher order operator $\Omega : (V \times \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}) \to (V \times \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0})$ where*

$$\begin{aligned}\Omega(F)(v,b) = {} & \int_0^b \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\, t} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, F(v', b-t)\, \mathrm{d}t \\ & + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) \right) \left(1 - \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} \cdot b} \right) + \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}} \cdot b} \end{aligned} \tag{6.1a}$$

when $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$, and for $v \in V_{\mathrm{P}}$:

$$\Omega(F)(v,b) = \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v')\, F(v', b) \Big) \tag{6.1b}$$

Proof. By setting $\varrho = \mathbb{1}_{T_{\mathrm{M}}}$, $V_{\mathrm{rc}} = V_{\mathrm{M}}$ and $V_{\mathrm{rp}} = V_{\mathrm{P}}$ in Thm. 5.2. □

We can indeed bring all the results established for the optimal ERR into the time-bounded setting. It is, in particular, possible to adapt the upper bound for the optimal ERR given in Cor. 4.23.1, the recursion of the optimal n-step ERRs given in Lem. 5.1 and splitting the resource bound described in Lem. 5.3 to this case. We can also conclude that the optimal ETR as a function of time-bound is Lipschitz continuous with the adapted constant $L = \bar{\rho}_{\mathrm{t}} + \bar{\lambda}(\bar{\rho}_{\mathrm{i}} + r_0 + \bar{\rho}_{\mathrm{f}})$, according to Thm. 5.7.

We have introduced the optimal ETR, not only because it represents a useful class of analyses, but also because its computation provides the basis for the computation of the more general class, namely the optimal ERR. The details are discussed in Sec. 6.4.

6.2 Discretisation

In this section we describe how to compute the optimal ETR via discretisation of the fixed point characterisation introduced in Thm. 6.2. It is not generally feasible to directly solve the fixed point characterisation due to the presence of the complex Volterra integral equations therein. The discretisation technique proposed here is the solution we have devised to tackle the problem. It yields a stable numerical algorithm with strict error bound. Moreover, it can be harvested to compute the optimal ERR through an efficient transformation that will be described in Sec. 6.4.

The general idea behind discretisation is to split the time interval from time zero up to the time bound into a finite number of equally sized chunks. This way we evaluate the optimal ETR at finitely many points, i. e. at the boundaries of each chunk. However, this does not necessarily decrease the complexity of our analysis. This is because the question how the model behaves inside each chunk, in general, can be highly complicated to answer and algorithmically intractable to analyse. Two observations help us to further simplify the behaviour of an MRA within each chunk. They are, firstly, “with high probability at most one Markovian transition is executed within each chunk” and secondly, “the reward gained after execution of the first Markovian transition is negligible”. The observations are in fact valid especially when the chunk length is small. The *discretised* optimal ETR is emerged from taking these observations into account while computing the optimal ETR. In practice, the discretised optimal ETR can be computed efficiently in polynomial time. In the sequel we show, in more details, how we utilise the observations to define the discretised optimal ETR. Moreover, we indicate that the error arising from the discretisation can be strictly bounded.

6.2.1 Discretised optimal ETR

Our goal is to approximately determine the optimal ETR, under a given time bound $b \in \mathbb{R}_{\geq 0}$ using the discretisation just discussed. We have defined the optimal ERR in Ch. 4 as the limit of *optimal n-step ERRs*. A similar procedure is used here to define the discretised optimal ETR, namely we define it as the limit of discretised optimal n-step ETRs. To do this, we first need to introduce the latter.

Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an MRA. We aim to compute the discretised optimal ETR withing a given time bound $b \geq 0$. The discretisation is applied by splitting interval $[0, b]$ into $K > 0$ chunk(s), each with length $\delta = \frac{b}{K}$. The length of each chunk is also referred to as *discretisation constant*. The idea is to evaluate the objective at equidistant time points $i\delta$, $i = 0, \ldots, K$ under two simplifying assumptions: at most one Markovian jump lies in each chunk $[j\delta, (j+1)\delta]$, $j = 0, \ldots, K-1$, and the reward obtained after this jump – if there is any – is ignored. The assumptions are inspired by the observations discussed before.

The assumptions simplify the computation of the optimal ETR, in particular for Markovian states. For $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$ it follows from Lem. 5.3 that for $n \in \mathbb{N}$ and $k > 0$

$$\begin{aligned}\overline{\underline{\mathbf{R}}}_{n+1}(v, k\delta) = {} & \overbrace{\int_0^{\delta} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_{n}(v', k\delta - t)\, \mathrm{d}t}^{(a)} \\ & + \underbrace{\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)}_{(b)} + \underbrace{\overline{\underline{\mathbf{R}}}_{n+1}(v, (k-1)\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}}_{(c)}\end{aligned}$$

Notice the difference between two indices n and k. The former counts the number of transitions whereas the latter refers to the chunks. The source of partial rewards (a), (b) and (c) becomes clear only in combination with the expectation splitting used in the proof of Lem. 5.1 together with time horizon splitting employed by the discretisation. To explain it more precisely we need to partition the set of paths starting from v according to the execution time of the first transition, which is $\check{\gamma}$. Put $J^{v}_{\leq\delta} := \{\check{\gamma} \xrightarrow{t} \pi \mid 0 \leq t \leq \delta, \pi \in \mathbb{P}\}$ and $J^{v}_{>\delta} := \{\check{\gamma} \xrightarrow{t} \pi \mid t > \delta, \pi \in \mathbb{P}\}$ as the set of paths staying less and respectively more than δ time units in v. We then split the reward gained by each partition separately. That is to say, we decompose the reward of each path in $J^{v}_{\leq\delta}$ into the reward gained before and after leaving v. Moreover, we break the reward of each path in $J^{v}_{>\delta}$ into the reward gained within the first chunk (inside interval $[0, \delta]$) the one gained thereafter. We accordingly write

$$\begin{aligned} R^{b}_{n+1}(\pi) &= \overbrace{\rho_{\mathrm{t}}(\check{\gamma})\, t + \rho_{\mathrm{i}}(\check{\gamma})}^{(b)} + \overbrace{R^{b-t}_{n}(\pi')}^{(a)} \qquad && J^{v}_{\leq\delta} \ni \pi = \check{\gamma} \xrightarrow{t} \pi' \\ R^{b}_{n+1}(\pi) &= \underbrace{\rho_{\mathrm{t}}(\check{\gamma})\, \delta}_{(b)} + \underbrace{R^{b-\delta}_{n}(\check{\gamma} \xrightarrow{t-\delta} \pi')}_{(c)} \qquad && J^{v}_{>\delta} \ni \pi = \check{\gamma} \xrightarrow{t} \pi' \end{aligned}$$

which also associates each of the partial rewards (a), (b) and (c) with its corresponding reward source. We can then conclude that the optimal $(n+1)$-step ETR can be split into three portions, coming from

- the reward gained after leaving v from paths in $J^{v}_{\leq\delta}$, denoted by (a),
- the reward gained before leaving v within the first chunk, denoted by (b), and
- the reward gained after the first δ time units from paths in $J^{v}_{>\delta}$, denoted by (c).

With the source of each partial reward being clear we can apply the simplifying assumptions suggested by the discretisation. Considering the discretisation step from time point $k\delta$ to $(k-1)\delta$, it is not hard to see that the assumptions only affect the computation of part (a). That involves the integral equation, which is also the most difficult part of the computation. From the assumptions, there will be no transition execution and no more reward after leaving v. To apply the assumptions, we simply replace $\overline{\underline{\mathbf{R}}}_n(v', k\delta - t)$ by $\overline{\underline{\mathbf{R}}}_n(v', (k-1)\delta)$ in part (a). This change brings both assumptions into play as it forces to stay at v' after leaving v until the end of discretisation step while collecting no reward. This then simplifies part (a) by

$$\int_0^\delta \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n(v', (k-1)\delta)\, \mathrm{d}t = \big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, \delta}\big) \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n(v', (k-1)\delta)$$

This result is an essential step towards the definition of the discretised optimal ETR.

We now define discretised optimal n-step ETRs using the simplification just discussed. They are specified by $\overline{\underline{\mathbf{R}}}_n^\delta : V \times \mathbb{N} \to \overline{\mathbb{R}}_{\geq 0}$ for a discretisation constant δ and $n \in \mathbb{N}$. Intuitively speaking, $\overline{\underline{\mathbf{R}}}_n^\delta(v, k)$ denotes the discretised optimal n-step ETR from state v within time bound $k\delta$. It can be seen as an approximation of $\overline{\underline{\mathbf{R}}}_n(v, k\delta)$. The next definition describes it formally in a recursive way.

Definition 6.3 (Discretised optimal n-step ETRs)**.** *Given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_\mathrm{M}}, \rho)$ and a discretisation constant $\delta > 0$. Then $\overline{\underline{\mathbf{R}}}_n^\delta : V \times \mathbb{N} \to \overline{\mathbb{R}}_{\geq 0}$ for $n \in \mathbb{N}$ is defined for boundary points as $\overline{\underline{\mathbf{R}}}_0^\delta := \mathbf{0}$ and $\overline{\underline{\mathbf{R}}}_n^\delta(v, 0) = \rho_\mathrm{f}(\check{\gamma})$ for $n > 0$ and $v \in V_\mathrm{M}$ with $T(v) = \{\check{\gamma}\}$. Furthermore, it holds for $n \in \mathbb{N}$ that*

$$\begin{aligned}\overline{\underline{\mathbf{R}}}_{n+1}^\delta(v, k) = {} & \big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, \delta}\big) \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}_n^\delta(v', k-1) \\ & + \Big(\frac{\rho_\mathrm{t}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_\mathrm{i}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, \delta}\Big) + \overline{\underline{\mathbf{R}}}_{n+1}^\delta(v, k-1)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, \delta} \qquad (6.2a)\end{aligned}$$

for $k > 0$ and $v \in V_\mathrm{M}$ with $T(v) = \{\check{\gamma}\}$;

$$\overline{\underline{\mathbf{R}}}_{n+1}^\delta(v, k) = \underset{\gamma \in T(v)}{\mathrm{mix}} \Big(\rho_\mathrm{i}(\gamma) + \sum_{v' \in V} \mathrm{d}_\gamma(v') \cdot \overline{\underline{\mathbf{R}}}_n^\delta(v', k)\Big) \qquad (6.2b)$$

for $k \in \mathbb{N}$ and $v \in V_\mathrm{P}$.

The recursion suggested by Eq. (6.2b) resembles the corresponding one for $\overline{\underline{\mathbf{R}}}_n$ given in Eq. (5.3c) on p. 78. As a consequence the reward coming from probabilistic transitions is computed in the same way with and without discretisation. Nevertheless, the discretisation employs the simplification as discussed previously in the computation for Markovian states in Eq. (6.2a). This is the situation where the computation is simplified.

In the next step, we define the discretised optimal ETR as the limit of the discretised optimal n-step ETRs when n goes to infinity. We shall first show that the limit exist by taking advantage of the fact that the discretised optimal n-step ETRs constitute an increasing sequence of functions. For that we provide another interpretation of Def. 6.3 and using an *order preserving operator* that transforms $\overline{\underline{\mathbf{R}}}^{\delta}_{n}$ into $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}$.

Definition 6.4. *For a given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$ we define operator $\overline{\underline{\Omega}}^{\delta}_{\rho} : (V \times \mathbb{N} \to \overline{\mathbb{R}}_{\geq 0}) \to (V \times \mathbb{N} \to \overline{\mathbb{R}}_{\geq 0})$ as, for $k \in \mathbb{N}$*

$$\begin{aligned}\overline{\underline{\Omega}}^{\delta}_{\rho}(F)(v,k) = {} & \Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathrm{r}_{\check{\gamma}} i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,F(v',k-i-1)\\ & + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}k\delta}\end{aligned}$$

when $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$ and

$$\overline{\underline{\Omega}}^{\delta}_{\rho}(F)(v,k) = \operatorname*{mix}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathrm{d}_{\gamma}(v')\,F(v',k)\Big)$$

for $v \in V_{\mathrm{P}}$. The min *and* max *operators are respectively taken for $\underline{\Omega}^{\delta}$ and $\overline{\Omega}^{\delta}$.*

It is possible to define the recursion described in Def. 6.3 using operator $\overline{\underline{\Omega}}^{\delta}_{\rho}$. We show that the recursion can be done by applying the operator, i. e. $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1} = \overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n})$. This fact is obvious for probabilistic states, but it needs to be elaborated for Markovian states. It is also worth noting that this case is the discrete version of Eq. (6.1a) where the integral is approximately computed by summation. We summarise the result here.

Lemma 6.5. *For a given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$ it holds that $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1} = \overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n})$, for $n \in \mathbb{N}$.*

Proof. We prove the lemma for all different cases described in Def. 6.3. The claim for $v \in V_{\mathrm{P}}$ follows directly from Eq. (6.2b) and Def. 6.4. Now assume $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$ and fix some $n \in \mathbb{N}$. For this case we shall show by induction on k that $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k) = \overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n})(v,k)$. First observe that for the induction base, $k = 0$, the claim follows from Def. 6.4 as $\overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n})(v,0) = \rho_{\mathrm{f}}(\check{\gamma}) = \overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,0)$. By induction hypothesis, assume that the claim holds for some $k \in \mathbb{N}$. We show then it holds for $k+1$. We use Eq. (6.2a) to obtain

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v,k+1) &= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta}\Big)\Big(\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k)+\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\\
&\quad+\underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v,k)\cdot\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta}\Big)\Big(\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k)+\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\\
&\quad+\underline{\overline{\Omega}}^{\delta}_{\rho}\Big(\underline{\overline{\mathbf{R}}}^{\delta}_{n}\Big)(v,k)\cdot\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta} && \Big[*\text{ I. H. }*\Big]\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta}\Big)\Big(\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k)+\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\\
&\quad+\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1)\\
&\quad+\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta}\Big)\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k)\\
&\quad+\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(i+1)\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1)\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta}\Big)\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k)\\
&\quad+\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{j=1}^{k}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}j\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-j)\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{j=0}^{k}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}j\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-j)\\
&\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}(k+1)\delta}\\
&= \underline{\overline{\Omega}}^{\delta}_{\rho}\Big(\underline{\overline{\mathbf{R}}}^{\delta}_{n}\Big)(v,k+1)
\end{aligned}$$

□

Lem. 6.5 brings yet another intuition how the discretisation works. It presents the discretisation as a technique for computing the integral involved in Eq. (6.1a).

For $v \in V_{\mathrm{M}}$ it gives for $k \in \mathbb{N}$ that

$$\begin{aligned}\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k) = \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1)\\ +\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\end{aligned}$$

On the other hand, it follows from Eq. (6.1a) in Thm. 6.2 that

$$\begin{aligned}\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta) = \int_{0}^{k\delta}\mathsf{r}_{\check{\gamma}}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_{n}(v',k\delta-t)\,\mathrm{d}t\\ +\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\end{aligned}$$

Comparing both equations reveals that the discretisation employs a technique similar to Riemann summation for estimation of the integral involved in the computation. It estimates $\overline{\underline{\mathbf{R}}}_{n}(v',\cdot)$ using a piecewise constant function that gives a constant value inside each of the k chunk(s) made by the discretisation. More precisely, $\overline{\underline{\mathbf{R}}}_{n}(v',t)$ for $j\delta \leq t < (j+1)\delta$, $j = 0,\ldots,k-1$ is approximated by $\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',j)$. We show in the next section that the error arising from this approximation is bounded.

The main reason for the representation of $\overline{\underline{\mathbf{R}}}^{\delta}_{n}$ using operator $\overline{\underline{\Omega}}^{\delta}_{\rho}$ is because the operator has a useful feature, namely it is *order preserving*. The next lemma indicates this property.

Lemma 6.6 (Order preservation). *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an MRA and $F, G : V \times \mathbb{N} \to \overline{\mathbb{R}}_{\geq 0}$ such that $F \leq G$, i. e. $G(v,k) \leq F(v,k)$ for every $v \in V$ and $k \in \mathbb{N}$. Then, for all $\delta > 0$ it holds that $\overline{\underline{\Omega}}^{\delta}_{\rho}(F) \leq \overline{\underline{\Omega}}^{\delta}_{\rho}(G)$.*

Proof. We need to show that $\overline{\underline{\Omega}}^{\delta}_{\rho}(F)(v,k) \leq \overline{\underline{\Omega}}^{\delta}_{\rho}(G)(v,k)$ for every $v \in V$ and $k \in \mathbb{N}$. For this we distinguish between two cases. For $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$, it holds that

$$\begin{aligned}\overline{\underline{\Omega}}^{\delta}_{\rho}(F)(v,k) &= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,F(v',k-i-1)\\ &\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\\ &\leq \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,G(v',k-i-1)\\ &\quad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\\ &= \overline{\underline{\Omega}}^{\delta}_{\rho}(G)(v,k)\end{aligned}$$

And similarly for $v \in V_{\mathrm{P}}$ we have

$$\begin{aligned}\overline{\underline{\Omega}}^{\delta}_{\rho}(F)(v,k) &= \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v')\, F(v',k)\Big) \\ &\leq \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v')\, G(v',k)\Big) \\ &= \overline{\underline{\Omega}}^{\delta}_{\rho}(G)(v,k)\end{aligned}$$

□

Being order preserving brings important properties. This is in particular helpful for proving that the sequence $\{\overline{\underline{\mathbf{R}}}^{\delta}_{n}\}_{n \in \mathbb{N}}$ is monotonically increasing and therefore its limit exists. The result is described in the next lemma.

Lemma 6.7. *For a given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$ it holds for all $n \in \mathbb{N}$ that $\overline{\underline{\mathbf{R}}}^{\delta}_{n} \leq \overline{\underline{\mathbf{R}}}^{\delta}_{n+1}$ and $\lim_{n \to \infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n}(v,k)$ exists for all $v \in V$ and $k \in \mathbb{N}$.*

Proof. First we prove by induction on n that $\overline{\underline{\mathbf{R}}}^{\delta}_{n} \leq \overline{\underline{\mathbf{R}}}^{\delta}_{n+1}$. The base case, $\overline{\underline{\mathbf{R}}}^{\delta}_{0} \leq \overline{\underline{\mathbf{R}}}^{\delta}_{1}$ is trivial since $\overline{\underline{\mathbf{R}}}^{\delta}_{0}$ is constantly zero. Now assume, for some $n \in \mathbb{N}$, it holds that $\overline{\underline{\mathbf{R}}}^{\delta}_{n} \leq \overline{\underline{\mathbf{R}}}^{\delta}_{n+1}$. By Lem. 6.6 we know that $\overline{\underline{\Omega}}^{\delta}_{\rho}$ is order preserving. It subsequently implies that $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1} = \overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n}) \leq \overline{\underline{\Omega}}^{\delta}_{\rho}(\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}) = \overline{\underline{\mathbf{R}}}^{\delta}_{n+2}$. The existence of the limit is a direct consequence of $\{\overline{\underline{\mathbf{R}}}^{\delta}_{n}\}_{n \in \mathbb{N}}$ being monotonically increasing. □

The consequence of Lem. 6.7 completes the story of this subsection. Thus far, the optimal discretised ETR has not been formally defined. Here we propose it as the limit of the discretised optimal n-step ETRs.

Definition 6.8 (Discretised optimal ETR)**.** *For a given MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ and discretisation constant $\delta > 0$, the* discretised optimal ETR, *$\overline{\underline{\mathbf{R}}}^{\delta}$ is defined for every $v \in V$ and $k \in \mathbb{N}$ as $\overline{\underline{\mathbf{R}}}^{\delta}(v,k) := \lim_{n \to \infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n}(v,k)$.*

We have thus far seen similar procedures in introducing concepts for the optimal ETR and its discrete version. For each of them we started from the definitions of the corresponding n-step functions. We showed that each of them constitutes an increasing sequence of functions whose limit then exists. And finally we defined the main objective as the limit of the corresponding sequence. We have proposed a characterisation of the optimal ETR in Thm. 6.2 as a fixed point. The corresponding characterisation does not yet exist for the discrete version. Here we continue the procedure and propose a fixed point characterisation for the discretised optimal ETR.

Theorem 6.9. *Given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$. Then $\overline{\underline{\mathbf{R}}}^{\delta}$ is the least fixed point of $\overline{\underline{\Omega}}^{\delta}_{\rho}$.*

Proof. We first prove that $\overline{\underline{\mathbf{R}}}^{\delta}$ is a fixed point of $\overline{\underline{\Omega}}^{\delta}_{\rho}$. For this it is enough to show that lim and $\overline{\underline{\Omega}}^{\delta}_{\rho}$ are interchangeable. More precisely, it suffices to confirm the correctness $(*)$ in the following equation.

$$\overline{\underline{\mathbf{R}}}^{\delta} = \lim_{n\to\infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n+1} = \lim_{n\to\infty} \overline{\underline{\Omega}}^{\delta}_{\rho}\big(\overline{\underline{\mathbf{R}}}^{\delta}_{n}\big) \overset{(*)}{=} \overline{\underline{\Omega}}^{\delta}_{\rho}\big(\lim_{n\to\infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n}\big) = \overline{\underline{\Omega}}^{\delta}_{\rho}\big(\overline{\underline{\mathbf{R}}}^{\delta}\big)$$

We inspect it for different kinds of states. For $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$, it is enough to show, for all $k \in \mathbb{N}$ that

$$\lim_{n\to\infty}\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1) =$$
$$\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\lim_{n\to\infty}\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1)$$

The above equation can be justified immediately by the monotone convergence theorem (Thm. 2.19) since sequence $\{\overline{\underline{\mathbf{R}}}^{\delta}_{n}\}_{n\in\mathbb{N}}$ is increasing.

Now we elaborate on the correctness of $(*)$ for $v \in V_{\mathrm{P}}$. For $\gamma \in T(v)$, put

$$p^{v,k}_{n}(\gamma) := \rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V}\mathrm{d}_{\gamma}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k)$$
$$p^{v,k}(\gamma) := \rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V}\mathrm{d}_{\gamma}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}(v',k)$$

As sequence $\{\overline{\underline{\mathbf{R}}}^{\delta}_{n}\}_{n\in\mathbb{N}}$ is increasing, it is not hard to see by the monotone convergence theorem that $\lim_{n\to\infty} p^{v,k}_{n} = p^{v,k}$. Thanks to the finiteness of T, we can further deduce that $\{p^{v,k}_{n}\}_{n\in\mathbb{N}}$ converges uniformly to $p^{v,k}$. Hence, Thm. 2.22 allows us to write

$$\lim_{n\to\infty}\ \operatorname*{mix}_{\gamma\in T(v)} p^{v,k}_{n}(\gamma) = \operatorname*{mix}_{\gamma\in T(v)} p^{v,k}(\gamma)$$

which completes the first part of the proof.

It remains to show that $\overline{\underline{\mathbf{R}}}^{\delta}$ is the least fixed point of $\overline{\underline{\Omega}}^{\delta}_{\rho}$. Take any fixed point $F : V \times \mathbb{N} \to \overline{\mathbb{R}}$ of $\overline{\underline{\Omega}}^{\delta}_{\rho}$, then $\overline{\underline{\mathbf{R}}}^{\delta}_{0} \leq F$ since $\overline{\underline{\mathbf{R}}}^{\delta}_{0}$ is constantly zero. By induction on n and using the fact that $\overline{\underline{\Omega}}^{\delta}_{\rho}$ is order preserving (Lem. 6.6) one can show that $\overline{\underline{\mathbf{R}}}^{\delta}_{n} \leq F$ for all $n \in \mathbb{N}$. Therefore, $\overline{\underline{\mathbf{R}}}^{\delta} = \lim_{n\to\infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n} \leq F$ and we are done. □

From algorithmic points of view, it is important to have a characterisation that is algorithmically tractable. We have proposed a characterisation of the optimal ETR in Thm. 6.2 as a fixed point. We have discussed however that the characterisation is not algorithmically tractable. This was the main argument for resorting to the discretisation approach. On the contrary, the characterisation given in the Thm. 6.9 directly proposes an algorithm for computing the discretised optimal ETR. In summary, Thm. 6.9 provides an effective and practical technique for our purpose. More precisely, it serves as a basis for developing an algorithm for computing the discretised optimal ETR.

The algorithm proceeds via computing the recursion suggested by operator $\overline{\underline{\Omega}}^{\delta}_{\rho}$. Given step $k \in \mathbb{N}$ it is straightforward to compute $\overline{\underline{\mathbf{R}}}^{\delta}(v,k)$ for $v \in V_{\mathrm{M}}$. For this we need to recursively compute the objective for all successors of v and all steps $\mathbb{N} \ni k' < k$ and then apply operator $\overline{\underline{\Omega}}^{\delta}_{\rho}$. Storing $\overline{\underline{\mathbf{R}}}^{\delta}(v',k')$ for all successor v' of v and all $k' < k$ might not be space efficient, specially if k is large. To tackle this problem, the solution established by the next lemma requires only to have the values for the $(k-1)$-th step.

Lemma 6.10. *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an MRA and $\delta > 0$ be a discretisation constant, then for every $v \in V_{\mathrm{M}}$ and every $k > 0$ it holds that*

$$\begin{aligned}\overline{\underline{\mathbf{R}}}^{\delta}(v,k) = \left(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\right)\sum_{v'\in V} \mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}(v',k-1)\\ +\left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\right)\left(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\right)+\overline{\underline{\mathbf{R}}}^{\delta}(v,k-1)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\end{aligned} \tag{6.3}$$

Proof. The proof is done via taking the limit of both sides of Eq. (6.2a) as n goes to infinity. After processing the left hand side, the claim follows from the monotone convergence theorem (since sequence $\overline{\underline{\mathbf{R}}}^{\delta}_{n}$ is monotone) and the fact that $\lim_{n\to\infty} \overline{\underline{\mathbf{R}}}^{\delta}_{n} = \overline{\underline{\mathbf{R}}}^{\delta}$ (Def. 6.8). □

Lem. 6.10 decreases the space complexity of computing the discretised optimal ETR for Markovian states. That is to say, if the vector of objective values for step $k-1$, i. e. $\overline{\underline{\mathbf{R}}}^{\delta}(\cdot,k-1)$ is ready, it is possible to computed $\overline{\underline{\mathbf{R}}}^{\delta}(v,k)$ for every $v \in V_{\mathrm{M}}$ using Eq. (6.3). For $v \in V_{\mathrm{P}}$, on the other hand, $\overline{\underline{\Omega}}^{\delta}_{\rho}$ suggests to compute a fixed point which is similar to the one used for computing the optimal reward in MDPs. This fixed point can be computed via *value* or *policy iteration* or solving the corresponding linear programming problem [For+11; Put05; BK08]. This completes the cycle of the algorithm for step k. We discuss about the algorithm and its complexity in more details in Sec. 6.3.2. Before that, we inspect in the next section how precise is to approximate the optimal ETR by its discrete version.

6.2.2 Error analysis

The whole idea of discretisation is relevant and useful only if it is sound, meaning that the error arising from the approximation vanishes when the discretisation constant tends to zero. The main topic of this section is to establish the soundness of the discretisation. We also establish a lower and an upper bound on the discretisation error. We use the bounds later to figure out the discretisation constant in order to guarantee a predefined accuracy level. This is of course only applicable when the optimal ETR is bounded. Therefore, for analysing the discretisation error, we only consider *NRC models*, for which the optimal ETR is *finite*. We deal with the NRD models later in the next section.

We discussed two main assumptions made by the discretisation in the previous section. They echo the observations that have been mentioned in the beginning of Sec. 6.2. We recap the observations here and study their likelihood when the discretisation constant tends to zero. This study intuitively clarifies the soundness of the discretisation. Let $\delta > 0$ be the discretisation constant, i. e. the length of each chunk made by the discretisation, then we observe that

1. with high probability each chunk carries at most one Markovian jump,

2. the expected reward gained after the first jump is negligible.

Now, let $\bar{\lambda}$ be the largest exit rate appearing in the model. Then, $1 - \mathrm{e}^{-\bar{\lambda}\delta}(1 + \bar{\lambda}\delta)$ is the maximum probability to have more than one Markovian jump in a chunk. This probability vanishes as δ goes to zero, thereby confirming the first observation. Since the model we analyse at the moment is NRC, the maximum reward it can gain (or lose) in each chunk is bounded by $L\delta$, where L is the Lipschitz constant associated with the model. Therefore the reward obtained after the first jump in each chunk is bounded above by $(1 - \mathrm{e}^{-\bar{\lambda}\delta})L\delta$, where the first term indicates the maximum probability of jumping within each chunk. When δ approaches zero, this reward tends towards zero as well. This justifies the second observation.

The above arguments intuitively explain why the discretisation is sound. They support the fact that the terms dropped out of the computation vanishes as the discretisation constant becomes smaller and smaller. Or equivalently, the approximation approaches the real objective as δ tends zero. In the remainder of this section we try to formally prove that statement.

We have seen in Ch. 4 that breaking the optimal ERR into partial rewards provides a useful tool to better understand the nature of reward acquisition. It helps us to figure out how the partial rewards influence the value of the optimal ERR. This has been used heavily to find out the upper bound for the optimal ERR proposed in Sec. 4.3.1 and to show that it is Lipschitz continuous in Sec. 5.2. Here we proceed in the same way for the *discretised optimal ETR*. We decompose the objective into partial rewards and then study their properties in order to establish particular inequalities that are helpful for proving the discretisation error.

Similar to the decomposition proposed for the optimal ERR by Def. 5.4, we decompose the discretised optimal ETR into two different kinds of partial rewards: *cumulative* and*final* rewards. As the names suggest cumulative rewards are gained all the way through the execution of transitions in the model whereas final rewards are granted at the moment the deadline is met. For a reward structure $\rho = (\rho_{\mathrm{t}}, \rho_{\mathrm{i}}, \rho_{\mathrm{f}})$, it is therefore the case that the source of cumulative rewards comes from ρ_{t} and ρ_{i}, and that of final rewards come from ρ_{f}.

Definition 6.11 (Discretised partial rewards)**.** *Given an MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_\mathrm{M}}, \rho)$ and discretisation constant $\delta > 0$, we build reward structures $\rho^{(\mathrm{c})} = (\rho_\mathrm{t}, \rho_\mathrm{i}, \mathbf{0})$ and $\rho^{(\mathrm{f})} = (\mathbf{0}, \mathbf{0}, \rho_\mathrm{f})$ upon ρ. Moreover, for $n \in \mathbb{N}$ let $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}$ and $\overline{\underline{\mathbf{R}}}_n^{\mathrm{f},\delta}$ be the discretised optimal n-step ETR under reward structures $\rho^{(\mathrm{c})}$ and $\rho^{(\mathrm{f})}$, respectively, as described in Def. 6.3. Finally,* the discretised partial rewards *are defined as $\overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta} := \lim_{n\to\infty} \overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}$ and $\overline{\underline{\mathbf{R}}}^{\mathrm{f},\delta} := \lim_{n\to\infty} \overline{\underline{\mathbf{R}}}_n^{\mathrm{f},\delta}$.*

The reward structure $\rho^{(\mathrm{c})}$ and $\rho^{(\mathrm{f})}$ are the projection of ρ into its first two and, respectively, its last component. They are therefore special kinds of reward structures. In this view the discretised partial rewards are also the special cases of the discretised optimal ETR. Hence, all of the results established for the discretised optimal ETR holds also for its partial versions.

Remark 6.12. *Being special cases of $\overline{\underline{\mathbf{R}}}_n^{\delta}$ means that $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}$ and $\overline{\underline{\mathbf{R}}}_n^{\mathrm{f},\delta}$ constitute increasing sequences of functions. Therefore, both $\lim_{n\to\infty} \overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}$ and $\lim_{n\to\infty} \overline{\underline{\mathbf{R}}}_n^{\mathrm{f},\delta}$ introduced by Def. 6.11 exist. The same is true for $\overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}$ and $\overline{\underline{\mathbf{R}}}^{\mathrm{f},\delta}$. In particular, they are the least fixed points of $\overline{\underline{\Omega}}_{\rho^{(\mathrm{c})}}^{\delta}$ and $\overline{\underline{\Omega}}_{\rho^{(\mathrm{f})}}^{\delta}$, respectively.*

We utilise certain properties of the discretised partial rewards and establish two inequalities that are useful for proving the discretisation error. In particular, we make use of the fact that cumulative rewards are increasing with respect to the time bound, in both continuous and discrete domains. In other words, the value of (discretised) cumulative rewards increases when the time bound is extended. This is simply because by extending the deadline either more rewards are gained or, in the worst case, the reward value stays the same. This property in general holds for nonnegative rewards that are collected cumulatively. It is though not true for final rewards. Assume, for instance, the model stays at a state with very large final reward, then extending the deadline corresponds to having higher probability to leave the state and therefore losing some rewards in expectation. Altogether, we show the following inequalities.

Lemma 6.13. *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_\mathrm{M}}, \rho)$ be an MRA and $\delta > 0$ be a discretisation constant. Then it holds for every $n, k \in \mathbb{N}$ and every $v \in V$ that*

$$\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}(v,k) \leq \overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,k\delta) \tag{6.4}$$

$$\overline{\underline{\mathbf{R}}}_n^{\delta} \leq \overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta} + \overline{\underline{\mathbf{R}}}_n^{\mathrm{f},\delta} \tag{6.5}$$

Proof. To prove Eq. (6.4) we first show that $\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,\cdot) : \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}$ is an increasing function. Fix any $n \in \mathbb{N}$, then we have $R_n^{\mathrm{c},b}(\pi) \leq R_n^{\mathrm{c},b'}(\pi)$ for all $\pi \in \mathbb{P}$ and $b \leq b'$. This yields $\mathbf{R}_n^{\mathrm{c}}(v,\sigma,b) \leq \mathbf{R}_n^{\mathrm{c}}(v,\sigma,b')$ for every $v \in V$, which consequently confirms for every $n \in \mathbb{N}$ and every $v \in V$ that

$$\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,b) \leq \overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,b') \qquad \left(b, b' \in \mathbb{R}_{\geq 0}, b \leq b'\right) \tag{6.6}$$

Now we proceed by induction on n and show Eq. (6.4) holds. The base case $n = 0$ is obvious, since $\overline{\underline{\mathbf{R}}}_0^{\mathrm{c},\delta}$ and $\overline{\underline{\mathbf{R}}}_0^{\mathrm{c}}$ are both constantly zero. By induction hypothesis, we assume for some $n \in \mathbb{N}$, every $v \in V$ and every $k \in \mathbb{N}$ that

$$\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}(v,k) \le \overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v,k\delta) \tag{6.7}$$

To prove the claim for $n+1$, we consider two cases. For $v \in V_{\mathrm{P}}$ we can write for every $k \in \mathbb{N}$ that

$$\begin{aligned}
\overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k) &= \operatorname*{mix}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V} \mathrm{d}_\gamma(v')\,\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}(v',k)\Big) && \big[\!*\text{ Eq. (6.2b) }*\big]\\
&\le \operatorname*{mix}_{\gamma\in T(v)}\Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v'\in V} \mathrm{d}_\gamma(v')\,\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v',k)\Big) && \big[\!*\text{ Eq. (6.7) }*\big]\\
&= \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c}}(v,k) && \big[\!*\text{ Eq. (5.3c) }*\big]
\end{aligned}$$

To prove the induction step for $v \in V_{\mathrm{M}}$, we apply another induction, this time on k. The claim for $k = 0$ holds since $\overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,0) = \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c}}(v,0) = \rho_{\mathrm{f}}(\check{\gamma})$ for all $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$. By induction hypothesis, suppose that for some $k \in \mathbb{N}$ and every $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$ we have

$$\overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k) \le \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c}}(v,k\delta) \tag{6.8}$$

We show the above inequality is correct for $k+1$ using the following derivations:

$$\begin{aligned}
&\overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k+1)\\
&\quad= \Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}(v',k) + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)\\
&\qquad + \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta} && \big[\!*\text{ Eq. (6.2a) }*\big]\\
&\quad= \int_0^\delta \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n^{\mathrm{c},\delta}(v',k)\,\mathrm{d}t + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)\\
&\qquad + \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\\
&\quad\le \int_0^\delta \mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v',k\delta)\,\mathrm{d}t + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)\\
&\qquad + \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta} && \big[\!*\text{ Eq. (6.7) }*\big]\\
&\quad\le \int_0^\delta \mathrm{r}_{\check{\gamma}}\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v',k\delta+\delta-t)\,\mathrm{d}t + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)\\
&\qquad + \overline{\underline{\mathbf{R}}}_{n+1}^{\mathrm{c},\delta}(v,k)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta} && \big[\!*\text{ Eq. (6.6) }*\big]\\
&\quad\le \int_0^\delta \mathrm{r}_{\check{\gamma}}\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\overline{\underline{\mathbf{R}}}_n^{\mathrm{c}}(v',k\delta+\delta-t)\,\mathrm{d}t + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\Big)
\end{aligned}$$

$$
\begin{aligned}
&+\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\,\delta} && \Big[\ast \text{ Eq. (6.8) } \ast\Big]\\
&=\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}\big(v,(k+1)\delta\big) && \Big[\ast \text{ Eq. (5.23) } \ast\Big]
\end{aligned}
$$

which completes the proof of Eq. (6.4).

We use induction once more to prove Eq. (6.5). The base case for $n=0$ is trivial. Suppose that Eq. (6.5) holds for some $n \in \mathbb{N}$. We show its correctness for $n+1$ by considering different cases. For $v \in V_{\mathrm{M}}$ we have

$$
\begin{aligned}
\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k) &= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k-i-1)\\
&\qquad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\\
&\qquad\qquad\Big[\ast \text{ Def. 6.4 and Lem. 6.5 } \ast\Big]\\
&\le \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\Big(\overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}_{n}(v',k-i-1)+\bar{\rho}_{\mathrm{f}}\Big)\\
&\qquad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta} \qquad \Big[\ast \text{ I. H. } \ast\Big]\\
&= \Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}i\delta}\sum_{v'\in V}\mathsf{d}_{\check{\gamma}}(v')\,\overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}_{n}(v',k-i-1)\\
&\qquad+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}+\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)\bar{\rho}_{\mathrm{f}}\\
&= \overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v,k)+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}+\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\Big)\bar{\rho}_{\mathrm{f}}\\
&\qquad\qquad\Big[\ast \text{ Def. 6.4 and Lem. 6.5 } \ast\Big]\\
&= \overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v,k)+\bar{\rho}_{\mathrm{f}}+\rho_{\mathrm{f}}(\check{\gamma})\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}k\delta}\\
&\le \overline{\underline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v,k)+\bar{\rho}_{\mathrm{f}} \qquad \Big[\ast\ \rho_{\mathrm{f}}(\check{\gamma})\le\bar{\rho}_{\mathrm{f}}\ \ast\Big]
\end{aligned}
$$

For $v \in V_{\mathrm{P}}$ we have

$$
\begin{aligned}
\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k) &= \underset{\gamma\in T(v)}{\mathrm{mix}}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\,(\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k)\Big) && \Big[\ast \text{ Eq. (6.2b) } \ast\Big]\\
&\le \underset{\gamma\in T(v)}{\mathrm{mix}}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\big(\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k)+\bar{\rho}_{\mathrm{f}}\big)\Big) && \Big[\ast \text{ I. H. } \ast\Big]\\
&= \underset{\gamma\in T(v)}{\mathrm{mix}}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k)+\bar{\rho}_{\mathrm{f}}\Big)\\
&= \underset{\gamma\in T(v)}{\mathrm{mix}}\Big(\rho_{\mathrm{i}}(\gamma)+\sum_{v'\in V}\mathsf{d}_{\gamma}(v')\overline{\underline{\mathbf{R}}}^{\delta}_{n}(v',k)\Big)+\bar{\rho}_{\mathrm{f}}\\
&= \overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k)+\bar{\rho}_{\mathrm{f}} && \Big[\ast \text{ Eq. (6.2b) } \ast\Big]
\end{aligned}
$$

□

This section aims to figure out a lower and an upper bound for the discretisation error. That is to say, we address the lower and the upper bound of the difference between the optimal ETR and its discrete approximation, defined by

$$\underline{\bar{d}}^{\delta}(v,k) := \underline{\overline{\mathbf{R}}}(v,k\delta) - \underline{\overline{\mathbf{R}}}^{\delta}(v,k) \tag{6.9}$$

where $\underline{\bar{d}}^{\delta}$ is referred to as *error function* for $\delta > 0$ and every $v \in V$ and $k \in \mathbb{N}$. As the common technique in this thesis, we first establish some appropriate bounds for the *n-step error function* described by $\underline{\bar{d}}^{\delta}_{n}(v,k) := \underline{\overline{\mathbf{R}}}_{n}(v,k\delta) - \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v,k)$ and then study its limit behaviour as n goes to infinity. For this purpose, we need an auxiliary inequality that deals with the lower and the upper bound of $\underline{\bar{\xi}}^{\delta,k-1}_{n}(v,\delta-t) := \underline{\overline{\mathbf{R}}}_{n}(v,k\delta-t) - \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v,k-1)$ for $0 \le t \le \delta$ and $k > 0$. Intuitively speaking, this is a *shifted form of the n-step error function* that reevaluates $\underline{\bar{d}}^{\delta}_{n}(v,k-1)$ in which the time bound of $\underline{\overline{\mathbf{R}}}_{n}$ is extended by $\delta - t$. It is formally represented by

$$\underline{\bar{\xi}}^{\delta,k}_{n}(v,\delta') := \underline{\overline{\mathbf{R}}}_{n}(v,k\delta+\delta') - \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v,k) \tag{6.10}$$

To see why we need to establish bounds on $\underline{\bar{\xi}}^{\delta,k-1}_{n}(v,\delta-t)$, we show how it appears in the computation of $\underline{\bar{d}}^{\delta}_{n+1}(v,k)$ for $v \in V_{\mathrm{M}}$. By Eq. (5.23) and (6.2a) we have

$$\begin{aligned}
\underline{\bar{d}}^{\delta}_{n+1}(v,k) &= \int_0^{\delta} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\overline{\mathbf{R}}}_{n}(v',k\delta-t)\,\mathrm{d}t \\
&\quad -\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big) \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-1) \\
&\quad + \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big(\underline{\overline{\mathbf{R}}}_{n+1}(v,(k-1)\delta) - \underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v,k-1)\Big) \\
&= \int_0^{\delta} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v') \Big(\underline{\overline{\mathbf{R}}}_{n}(v',k\delta-t) - \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v',k-1)\Big)\mathrm{d}t \\
&\quad + \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\, \underline{\bar{d}}^{\delta}_{n+1}(v,k-1) \\
&= \int_0^{\delta} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\bar{\xi}}^{\delta,k-1}_{n}(v,\delta-t)\,\mathrm{d}t + \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\, \underline{\bar{d}}^{\delta}_{n+1}(v,k-1)
\end{aligned} \tag{6.11}$$

In the above equation, observe that the term inside the integral contains $\underline{\bar{\xi}}^{\delta,k-1}_{n}(v,\delta-t)$. It is therefore helpful to know its bound while we inductively determine the bounds of $\underline{\bar{d}}^{\delta}_{n+1}$. This is addressed in the following two lemmas.

Lemma 6.14. *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an NRC MRA and $\delta > 0$ be a discretisation constant, then for a* Markovian state $v \in V_{\mathrm{M}}$ *it holds*

$$\underline{\bar{\xi}}^{\delta,k}_{n}(v,\delta') \le \delta' L^{\mathrm{c}} + \Big(k\delta L^{\mathrm{c}} + \bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \underline{\bar{d}}^{\delta}_{n}(v,k)\,\mathrm{e}^{-\bar{\lambda}\delta'} \tag{6.12}$$

$$\underline{\bar{\xi}}^{\delta,k}_{n}(v,\delta') \ge -\bar{\rho}_{\mathrm{f}} \cdot \Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \underline{\bar{d}}^{\delta}_{n}(v,k)\,\mathrm{e}^{-\bar{\lambda}\delta'} \tag{6.13}$$

where $L^{\mathrm{c}} = \bar{\rho}_{\mathrm{t}} + (\bar{\rho}_{\mathrm{i}} + r_0)\bar{\lambda}$ *and* $n, k \in \mathbb{N}$, $\delta' \in \mathbb{R}_{\geq 0}$.

Proof. Both inequalities hold for $n = 0$, as $\underline{\bar{\xi}}_0^{\delta,\cdot}$ and $\underline{\bar{d}}_0^{\delta}$ are constantly zero. It is therefore enough to show them for every nonzero natural n. We start with Eq. (6.13) and first prove a lower bound for $\underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta + \delta')$. Suppose $T(v) = \{\check{\gamma}\}$, then we consider $\underline{\overline{\mathbf{R}}}_n \geq \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_n$ and combine it with Eq. (5.23) to obtain

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta + \delta') \geq{}& \int_0^{\delta'} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v') \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_n(v', k\delta + \delta' - t)\, \mathrm{d}t \\
&+ \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'}\Big) + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \\
={}& \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v, k\delta + \delta') - \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \\
&\hfill \big[* \text{ Eq. (5.23) } *\big] \\
\geq{}& \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v, k\delta) - \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \\
&\hfill \big[* \text{ Eq. (6.6) } *\big] \\
={}& \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v, k\delta)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'}\Big) + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \\
\geq{}& \underline{\overline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v, k)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'}\Big) + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \quad \big[* \text{ Eq. (6.4) } *\big] \\
\geq{}& \underline{\overline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v, k)\Big(1 - \mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\bar{\lambda}\delta'}
\end{aligned} \tag{6.14}$$

where the last inequality follows from the fact that the right hand side is decreasing in $\bar{\lambda}$. This can be easily verified by taking the derivative of the right hand side with respect to $\bar{\lambda}$. The claim then follows by $\underline{\overline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1} \leq \underline{\overline{\mathbf{R}}}_{n+1}$. Now we establish an upper bound on $\underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k)$.

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k) ={}& \underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k)\Big(1 - \mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k)\, \mathrm{e}^{-\bar{\lambda}\delta'} \\
\leq{}& \Big(\underline{\overline{\mathbf{R}}}^{\mathrm{c},\delta}_{n+1}(v, k) + \bar{\rho}_{\mathrm{f}}\Big)\Big(1 - \mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k)\, \mathrm{e}^{-\bar{\lambda}\delta'}
\end{aligned} \tag{6.15}$$

where the last derivation follows from Eq. (6.5). After subtracting Eq. (**??**) from Eq. (6.14) the lower bound follows.

To obtain the upper bound we make use of the fact that the model is NRC. We then consider $\underline{\overline{\mathbf{R}}}_n \leq \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_n + \bar{\rho}_{\mathrm{f}}$ and combine it with Eq. (5.23) to obtain

$$\begin{aligned}
\underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta + \delta') \leq{}& \int_0^{\delta'} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\Big(\underline{\overline{\mathbf{R}}}^{\mathrm{c}}_n(v', k\delta + \delta' - t) + \bar{\rho}_{\mathrm{f}}\Big)\mathrm{d}t \\
&+ \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'}\Big) + \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta'} \\
={}& \int_0^{\delta'} \mathrm{r}_{\check{\gamma}}\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}} t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \underline{\overline{\mathbf{R}}}^{\mathrm{c}}_n(v', k\delta + \delta' - t)\, \mathrm{d}t
\end{aligned}$$

$$\begin{aligned}
&+\Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}}+\rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\Big)+\bar{\rho}_{\mathrm{f}}\cdot\big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\big)\\
&\quad+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\\
&=\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta+\delta')-\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}+\bar{\rho}_{\mathrm{f}}\cdot\big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\big)\\
&\quad+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'} && \big[*\text{ Eq. (5.23) }*\big]\\
&\le L^{\mathrm{c}}\delta'+\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta)-\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\\
&\quad+\bar{\rho}_{\mathrm{f}}\cdot\big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\big)+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'} && \big[*\text{ Lem. (5.6) }*\big]\\
&=L^{\mathrm{c}}\delta'+\Big(\overline{\underline{\mathbf{R}}}^{\mathrm{c}}_{n+1}(v,k\delta)+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\Big)\\
&\quad+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\\
&\le L^{\mathrm{c}}\delta'+\Big(k\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\Big)+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta'}\\
& && \big[*\text{ Cor. 4.23.1 }*\big]\\
&\le L^{\mathrm{c}}\delta'+\Big(k\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big)+\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\,\mathrm{e}^{-\bar{\lambda}\delta'} && (6.16)
\end{aligned}$$

Note that the last inequality follows from the fact that the right hand side is increasing in $\bar{\lambda}$. This can be confirmed by taking its derivative with respect to $\bar{\lambda}$. The derivative is positive since $\overline{\underline{\mathbf{R}}}_{n+1}(v,k\delta)\le k\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}$ by Cor. 4.23.1 as v is Markovian and thereby resource consuming. Finally, the upper bound follows after subtracting $\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k)\ge\overline{\underline{\mathbf{R}}}^{\delta}_{n+1}(v,k)\,\mathrm{e}^{-\bar{\lambda}\delta'}$ from Eq. (6.16). □

We remark on two points regarding Lem. 6.14. Firstly, the conclusion of the lemma is only valid for NRC models. The proof indeed makes use of Lipschitz continuity of NRC models. Secondly, we elaborate on L^{c}. In fact, it is the Lipschitz constant associated with the optimal cumulative ETR ($\overline{\underline{\mathbf{R}}}^{\mathrm{c}}$), i. e. when the final rewards are zero everywhere. Hence, it can be obtained by setting $\bar{\rho}_{\mathrm{f}}=0$ and $\underline{\varrho}=1$ in the Lipschitz constant provided by Lem. 5.6.

The above lemma relates the shifted error function with the error function for Markovian states in a recursive way. The following lemma however provides recursive inequalities between the shifted error function of a probabilistic state and that of its successors.

Lemma 6.15. *Let $\mathcal{R}:=(\mathcal{M},\mathbb{1}_{T_{\mathrm{M}}},\rho)$ be an MRA and $\delta>0$ be a discretisation constant, and suppose v is* probabilistic. *Then, for every $n,k\in\mathbb{N}$ and every $\delta'\in\mathbb{R}_{\ge0}$ there exist $\underline{\gamma},\bar{\gamma}\in T(v)$ such that*

$$\overline{\underline{\xi}}^{\delta,k}_{n+1}(v,\delta')\le\sum_{v'\in V}\mathrm{d}_{\overline{\underline{\gamma}}}(v')\,\overline{\underline{\xi}}^{\delta,k}_{n}(v',\delta') \tag{6.17}$$

Similarly for the lower bound, there exist $\underline{\gamma},\bar{\gamma}\in T(v)$ such that

$$\overline{\underline{\xi}}^{\delta,k}_{n+1}(v,\delta')\ge\sum_{v'\in V}\mathrm{d}_{\overline{\underline{\gamma}}}(v')\,\overline{\underline{\xi}}^{\delta,k}_{n}(v',\delta') \tag{6.18}$$

Proof. The proof is by construction, namely we show that such transitions exist for both the lower and the upper bound. We first look into the upper bound and put

$$
\begin{aligned}
\underline{\gamma} &= \operatorname*{arg\,min}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v') \underline{\mathbf{R}}^{\delta}_{n}(v', k) \Big) \\
\bar{\gamma} &= \operatorname*{arg\,max}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v') \overline{\mathbf{R}}_{n}(v', k\delta + \delta') \Big)
\end{aligned}
$$

We then combine Eq. (6.10) with (5.3c) to get

$$
\begin{aligned}
\underline{\bar{\xi}}^{\delta,k}_{n+1}(v, \delta') = \underline{\overline{\mathbf{R}}}_{n+1}(v, k\delta + \delta') - \underline{\overline{\mathbf{R}}}^{\delta}_{n+1}(v, k) \\
&= \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v')\, \underline{\overline{\mathbf{R}}}_{n}(v', k\delta + \delta') \Big) \\
&\quad - \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v')\, \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v', k) \Big) \\
&\leq \rho_{\mathrm{i}}(\underline{\bar{\gamma}}) + \sum_{v' \in V} \mathrm{d}_{\underline{\bar{\gamma}}}(v')\, \underline{\overline{\mathbf{R}}}_{n}(v', k\delta + \delta') \\
&\quad - \Big(\rho_{\mathrm{i}}(\underline{\bar{\gamma}}) + \sum_{v' \in V} \mathrm{d}_{\underline{\bar{\gamma}}}(v')\, \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v', k) \Big) \\
&= \sum_{v' \in V} \mathrm{d}_{\underline{\bar{\gamma}}}(v') \Big(\underline{\overline{\mathbf{R}}}_{n}(v', k\delta + \delta') - \underline{\overline{\mathbf{R}}}^{\delta}_{n}(v', k) \Big) \\
&= \sum_{v' \in V} \mathrm{d}_{\underline{\bar{\gamma}}}(v')\, \underline{\bar{\xi}}^{\delta,k}_{n}(v', \delta')
\end{aligned}
$$

For the lower bound it is enough to define

$$
\begin{aligned}
\underline{\gamma} &= \operatorname*{arg\,min}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \underline{\mathbf{R}}_{n}(v', k\delta + \delta') \Big) \\
\bar{\gamma} &= \operatorname*{arg\,max}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \overline{\mathbf{R}}^{\delta}_{n}(v', k) \Big)
\end{aligned}
$$

This follows then by substituting the '≤' with '≥' in the above steps. □

Lem. 6.15 implies that the lower and upper bounds of the shifted error function propagate by taking one probabilistic transitions. In particular, it allows such bounds from any probabilistic state are lifted from that of its successors. This also means that computation of the optimal ETR for probabilistic states does not arise any extra error. This result together with the result of Lem. 6.14 provides a useful tool for proving bounds on the discretisation error.

Before looking into the discretisation error, we need to define two functions that are used to determine the bounds for discretisation error. We assume NRC MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$ are given. Once more we distinguish between cumulative and final rewards. We use $\mathfrak{E}^{\mathrm{c}}_{\mathcal{R}}$ and $\mathfrak{E}^{\mathrm{f}}_{\mathcal{R}}$ to denote the upper bounds – to be established – on the discretisation error for cumulative and final rewards, respectively, where

$$\mathfrak{E}^{\mathrm{c}}_{\mathcal{R}}(k,\delta) = k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-i\delta\bar{\lambda}}\big(1+\bar{\lambda}\delta\big)^{i} \tag{6.19}$$

$$\mathfrak{E}^{\mathrm{f}}_{\mathcal{R}}(k,\delta) = \bar{\rho}_{\mathrm{f}}\cdot\Big(1-\mathrm{e}^{-k\delta\bar{\lambda}}\big(1+\bar{\lambda}\delta\big)^{k}\Big) \tag{6.20}$$

The MRA is often clear from the context, so we may drop it from the terms and simply write $\mathfrak{E}^{\mathrm{c}}$ and $\mathfrak{E}^{\mathrm{f}}$. Using the above functions we propose a lower and an upper bound for $\bar{\underline{\xi}}^{\delta,k}$ for $k \in \mathbb{N}$.

Lemma 6.16. *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an NRC MRA and $\delta > 0$ be a discretisation constant, then it holds for every $v \in V$ and every $\delta' \geq 0$ that*

$$\bar{\underline{\xi}}^{\delta,k}_{n}(v,\delta') \leq \delta' L + \Big(k\delta L + \bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \Big(\mathfrak{E}^{\mathrm{c}}(k,\delta)+\mathfrak{E}^{\mathrm{f}}(k,\delta)\Big)\,\mathrm{e}^{-\bar{\lambda}\delta'} \tag{6.21}$$

$$\bar{\underline{\xi}}^{\delta,k}_{n}(v,\delta') \geq -\bar{\rho}_{\mathrm{f}}\cdot\Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big) - \mathfrak{E}^{\mathrm{f}}(k,\delta)\,\mathrm{e}^{-\bar{\lambda}\delta'} \tag{6.22}$$

for all $n, k \in \mathbb{N}$.

Proof. We prove by induction on n that both Eq. (6.21) and (6.22) hold. The base case is obvious as $\bar{\underline{\xi}}^{\delta,k}_{0}$ is constantly zero for every $k \in \mathbb{N}$. We now assume that the claim holds for some $n \in \mathbb{N}$, and prove it also holds for $n+1$. We consider two cases: $v \in V_{\mathrm{M}}$ and $v \in V_{\mathrm{P}}$. The latter case can be simply addressed by Lem. 6.15. That is to say, there are $\underline{\gamma}, \bar{\gamma} \in T(v)$ such that

$$\begin{aligned}
\bar{\underline{\xi}}^{\delta,k}_{n+1}(v,\delta') &\leq \sum_{v'\in V}\mathrm{d}_{\underline{\bar{\gamma}}}(v')\,\bar{\underline{\xi}}^{\delta,k}_{n}(v',\delta') && \big[*\text{ Eq. (6.17) }*\big]\\
&\leq \delta' L + \Big(k\delta L + \bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}\delta'}\Big) + \Big(\mathfrak{E}^{\mathrm{c}}(k,\delta)+\mathfrak{E}^{\mathrm{f}}(k,\delta)\Big)\,\mathrm{e}^{-\bar{\lambda}\delta'}\\
& && \big[*\text{ I. H. }*\big]
\end{aligned}$$

The lower bound can be obtained analogously using Eq. (6.18).

We now consider $v \in V_{\mathrm{M}}$ and employ the result of Lem. 6.14. Accordingly it is enough to show that

$$-\mathfrak{E}^{\mathrm{f}}(k,\delta) \leq \bar{\underline{d}}^{\delta}_{n+1}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta)+\mathfrak{E}^{\mathrm{f}}(k,\delta) \tag{6.23}$$

We prove it by induction on k. In the base case, $k = 0$, all the terms involving in Eq. (6.23) are zero, as a result the inequality holds. Now, suppose Eq. (6.23) holds for some $k \in \mathbb{N}$, then we show that it also holds for $k+1$. First note by Eq. (6.11) that

$$\bar{\underline{d}}^{\delta}_{n+1}(v,k+1) = \int_{0}^{\delta}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\,\bar{\underline{\xi}}^{\delta,k}_{n}(v,\delta-t)\,\mathrm{d}t + \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\,\bar{\underline{d}}^{\delta}_{n+1}(v,k) \tag{6.24}$$

To obtain a lower and an upper bound for the above equation, we apply the hypothesis from the induction on “*n*” to obtain

$$
\begin{aligned}
&\bar{\underline{\xi}}^{\delta,k}_{n}(v,\delta-t)\\
&\quad\leq(\delta-t)L^{\mathrm{c}}+\Big(k\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}(\delta-t)}\Big)+\Big(\mathfrak{E}^{\mathrm{c}}(k,\delta)+\mathfrak{E}^{\mathrm{f}}(k,\delta)\Big)\,\mathrm{e}^{-\bar{\lambda}(\delta-t)}\\
&\quad=(\delta-t)L^{\mathrm{c}}+\Big(k\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\bar{\lambda}(\delta-t)}\Big)+\bar{\rho}_{\mathrm{f}}\cdot\Big(1-\mathrm{e}^{-k\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\Big)\,\mathrm{e}^{-\bar{\lambda}(\delta-t)}\\
&\qquad+\Big(k\delta L^{\mathrm{c}}-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i}\Big)\,\mathrm{e}^{-\bar{\lambda}(\delta-t)}\\
&\quad=(k+1)\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}-tL^{\mathrm{c}}-\frac{\mathrm{e}^{\bar{\lambda}t}L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}\\
&\qquad-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\mathrm{e}^{\bar{\lambda}t}
\end{aligned}
\tag{6.25}
$$

and similarly for the lower bound it gives

$$
\bar{\underline{\xi}}^{\delta,k}_{n}(v,\delta-t)\geq-\bar{\rho}_{\mathrm{f}}+\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\mathrm{e}^{\bar{\lambda}t}
\tag{6.26}
$$

Note that the lower bound is obtained by flipping the sign of the upper bound when $L^{\mathrm{c}}=0$. The proof of the lower bound can then be obtained by tweaking that of the upper bound. Therefore we first look into the upper bound. We substitute Eq. (6.25) into (6.24) and also applying the hypothesis of the induction on “*k*” to have

$$
\begin{aligned}
&\bar{\underline{d}}^{\delta}_{n+1}(v,k+1)\\
&\qquad\leq\int_{0}^{\delta}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\big((k+1)\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}-tL^{\mathrm{c}}\big)\,\mathrm{d}t\\
&\qquad\quad-\int_{0}^{\delta}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\Big(\frac{\mathrm{e}^{\bar{\lambda}t}L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}\Big)\,\mathrm{d}t\\
&\qquad\quad-\int_{0}^{\delta}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}t}\sum_{v'\in V}\mathrm{d}_{\check{\gamma}}(v')\big(\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\mathrm{e}^{\bar{\lambda}t}\big)\,\mathrm{d}t\\
&\qquad\quad+\Big(k\delta L^{\mathrm{c}}-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\sum_{i=0}^{k-1}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i}\Big)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\\
&\qquad\quad+\bar{\rho}_{\mathrm{f}}\cdot\Big(1-\mathrm{e}^{-k\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\Big)\,\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\\
&\qquad=\Big((k+1)\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}\Big)\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\Big)-\frac{L^{\mathrm{c}}}{\mathrm{r}_{\check{\gamma}}}\Big(1-\mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}(1+\mathrm{r}_{\check{\gamma}}\delta)\Big)\\
&\qquad\quad-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}\int_{0}^{\delta}\mathrm{r}_{\check{\gamma}}\,\mathrm{e}^{(\bar{\lambda}-\mathrm{r}_{\check{\gamma}})t}\,\mathrm{d}t
\end{aligned}
$$

$$\begin{aligned}
&-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\int_{0}^{\delta}\mathsf{r}_{\check{\gamma}}\,\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})t}\,\mathrm{d}t+k\delta L^{\mathrm{c}}\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\\
&-\frac{\mathrm{e}^{(\bar{\lambda}-\check{\gamma})\delta}L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}+\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\\
&-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})\delta}\\
={}&(k+1)\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}-\frac{L^{\mathrm{c}}}{\mathsf{r}_{\check{\gamma}}}\big(1-\mathrm{e}^{-\mathsf{r}_{\check{\gamma}}\delta}\big)\\
&-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}\Big(\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})\delta}+\int_{0}^{\delta}\mathsf{r}_{\check{\gamma}}\,\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})t}\,\mathrm{d}t\Big)\\
&-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}\Big(\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})\delta}+\int_{0}^{\delta}\mathsf{r}_{\check{\gamma}}\,\mathrm{e}^{(\bar{\lambda}-\mathsf{r}_{\check{\gamma}})t}\,\mathrm{d}t\Big)
\end{aligned}\tag{6.27}$$

We extract two functions from the above equation to determine how they vary. Put $F, G : (0,\bar{\lambda}] \to \mathbb{R}_{\geq 0}$ with

$$F(r) := \frac{L^{\mathrm{c}}}{r}\big(1-\mathrm{e}^{-r\delta}\big) \qquad \text{and} \qquad G(r) := \mathrm{e}^{(\bar{\lambda}-r)\delta}+\int_{0}^{\delta} r\mathrm{e}^{(\bar{\lambda}-r)t}\,\mathrm{d}t$$

As it can be seen, $F(\mathsf{r}_{\check{\gamma}})$ and $G(\mathsf{r}_{\check{\gamma}})$ both appears in the right hand side of the above equation. It is not hard to see that both functions are differentiable on $(0,\bar{\lambda})$, hence we have $F'(r) = -\frac{L^{\mathrm{c}}}{r^2}(1-\mathrm{e}^{-r\delta}(1+r\delta)) \leq 0$ and $G'(r) = -\int_0^{\delta}\bar{\lambda}t\mathrm{e}^{(\bar{\lambda}-r)t}\,\mathrm{d}t \leq 0$, where the former inequality comes from $\mathrm{e}^{x} \geq 1+x$ for $x \geq 0$. Since both functions are decreasing, we have

$$F(\mathsf{r}_{\check{\gamma}}) \geq \frac{L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big) = F(\bar{\lambda}) \qquad \text{and} \qquad G(\mathsf{r}_{\check{\gamma}}) \geq 1+\bar{\lambda}\delta = G(\bar{\lambda})$$

Including them into Eq. (6.27) gives

$$\begin{aligned}
\bar{\underline{d}}^{\delta}_{n+1}(v,k+1) &\leq (k+1)\delta L^{\mathrm{c}}+\bar{\rho}_{\mathrm{f}}-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\\
&\quad-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\sum_{i=1}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i-1}(1+\bar{\lambda}\delta)\\
&\quad-\bar{\rho}_{\mathrm{f}}\,\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k}(1+\bar{\lambda}\delta)\\
&= (k+1)\delta L^{\mathrm{c}}-\frac{L^{\mathrm{c}}}{\bar{\lambda}}\big(1-\mathrm{e}^{-\bar{\lambda}\delta}\big)\sum_{i=0}^{k}\mathrm{e}^{-i\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{i}\\
&\quad+\bar{\rho}_{\mathrm{f}}\big(1-\mathrm{e}^{-(k+1)\delta\bar{\lambda}}(1+\bar{\lambda}\delta)^{k+1}\big)\\
&= \mathfrak{E}^{\mathrm{c}}(k+1,\delta)+\mathfrak{E}^{\mathrm{f}}(k+1,\delta)
\end{aligned}$$

which completes the proof of the upper bound. That of the lower bound is essentially similar, by setting $L^{\mathrm{c}} = 0$ and flipping the signs and thereby the inequalities operators when necessary. □

We point out that Lem. 6.16 essentially provides bounds for the discretisation error. The bounds are simply obtained by setting $\delta' = 0$ in Eq. (6.21) and (6.22), as $\underline{\bar{d}}^{\delta}_{n}(v,k) = \underline{\bar{\xi}}^{\delta,k}_{n}(v,0)$ for every $v \in V$ and every $n,k \in \mathbb{N}$. There is a possibility to tighten the lower bound for special kinds of objectives. It happens in particular when the cumulative and final rewards are computed separately, for instance, for the optimal time-bounded reachability or the optimal time-bounded cumulative rewards. As such objectives are useful in industrial applications, it is worthwhile to establish a better bound for them, leading to more efficient computation. We formalise this improvement using a specifically structured MRA called *absorbing*, which is defined next.

Definition 6.17 (Absorbing MRA). *An MRA $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ is absorbing iff for all $\gamma \in T_{\mathrm{rc}}$, $\rho_{\mathrm{f}}(\gamma) > 0$ implies $\mathsf{d}_{\gamma} = \Delta_{\mathsf{v}_{\gamma}}$ with $\mathsf{r}_{\gamma} > 0$.*

Intuitively speaking, in an absorbing MRA every transition with a nonzero final reward is indeed a Markovian self loop. The states exhibiting such transitions are also called *absorbing states*. As a result, a path entering in an absorbing state stays there forever. This brings an important feature, namely it makes the total reward increasing with respect to increases in time bound. More precisely, by extending the deadline it cannot happen that the total reward decreases. This has a direct impact on the lower bound, formalised in the next lemma.

Lemma 6.18. *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an* absorbing *NRC MRA and δ be a discretisation constant, then it holds for all $v \in V$ that*

$$0 \leq \underline{\bar{d}}^{\delta}_{n}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta) + \mathfrak{E}^{\mathrm{f}}(k,\delta)$$

for all $n,k \in \mathbb{N}$.

Proof. The upper bound directly follows from Eq. (6.21) after setting $\delta' = 0$. For the lower bound we shall show that $\underline{\overline{\mathbf{R}}}^{\delta}_{n}(v,k) \leq \underline{\overline{\mathbf{R}}}_{n}(v,k\delta)$ for every v,k,n. The proof goes in the same line as that of Lem. 6.13. We first show that $\underline{\overline{\mathbf{R}}}_{n}(v,\cdot) : \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}$ is an increasing function. Fix any $n \in \mathbb{N}$, then we have $R^{b}_{n}(\pi) \leq R^{b'}_{n}(\pi)$ for all $\pi \in \mathbb{P}$ and $b \leq b'$. Note that this is not true in general since extending the deadline may reduce the final reward that is gained by the path. However, this situation cannot happen for an absorbing MRA since as soon as the path visits an absorbing state it cannot leave it. Therefore the final reward stays the same even though the deadline is extended. This is lifted up to the expectation and thereby to the optimal ETR, i. e. $\underline{\overline{\mathbf{R}}}_{n}(v,b) \leq \underline{\overline{\mathbf{R}}}_{n}(v,b')$. The rest of the proof is similar to the proof Eq. (6.4) in Lem. 6.13. To adapt the proof for this case, it is enough to replace $\underline{\overline{\mathbf{R}}}^{\mathrm{c}}$ with $\underline{\overline{\mathbf{R}}}$ and $\underline{\overline{\mathbf{R}}}^{\mathrm{c},\delta}$ with $\underline{\overline{\mathbf{R}}}^{\delta}$ there. □

We have thus far established the lower and the upper bound for n-step discretisation error for an arbitrary NRC MRA. If the model is absorbing as well, the lower bound is tightened by the above lemma. To propose a stable numerical algorithm for computing the optimal ETR, it is required to have a strict bound for the discretisation error. Since the bounds do not depend on n, we can lift the bounds to the error function. This is the main result of this section, formulated in the next theorem.

Theorem 6.19 (Discretisation error). *Let $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ be an NRC MRA and $\delta > 0$ be a discretisation constant, then it holds for all $v \in V$ that*

$$-\mathfrak{E}^{\mathrm{f}}(k,\delta) \leq \bar{\underline{d}}^{\delta}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta) + \mathfrak{E}^{\mathrm{f}}(k,\delta) \tag{6.28}$$

$$0 \leq \bar{\underline{d}}^{\delta}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta) + \mathfrak{E}^{\mathrm{f}}(k,\delta) \qquad \Big(\textit{when } \mathcal{R} \textit{ is absorbing}\Big) \tag{6.29}$$

for all $k \in \mathbb{N}$.

Proof. It holds by Lem. 6.16 and 6.18 that

$$-\mathfrak{E}^{\mathrm{f}}(k,\delta) \leq \bar{\underline{d}}_{n}^{\delta}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta) + \mathfrak{E}^{\mathrm{f}}(k,\delta)$$

$$0 \leq \bar{\underline{d}}_{n}^{\delta}(v,k) \leq \mathfrak{E}^{\mathrm{c}}(k,\delta) + \mathfrak{E}^{\mathrm{f}}(k,\delta) \qquad \Big(\text{when } \mathcal{R} \text{ is absorbing}\Big)$$

The claim simply follows by taking the limit of the above inequalities as n goes to infinity. Nevertheless it still needs to be confirmed that $\lim_{n\to\infty} \bar{\underline{d}}_{n}^{\delta} = \bar{\underline{d}}^{\delta}$. This is the case since the limits of $\underline{\overline{\mathbf{R}}}_{n}$ and $\underline{\overline{\mathbf{R}}}_{n}^{\delta}$ when n goes to infinity are both finite. The former follows from the fact that $\mathcal{R}$ is NRC and thereby the latter comes after taking the limit of the sequence of inequalities $\underline{\overline{\mathbf{R}}}_{n}^{\delta} \leq \underline{\overline{\mathbf{R}}}_{n}^{\mathrm{c}} + \bar{\rho}_{\mathrm{f}} \leq \underline{\overline{\mathbf{R}}}_{n} + \bar{\rho}_{\mathrm{f}}$. □

The theorem indicates that the discretisation scheme produces a strict error bound. By investigating the bounds it becomes clear that the smaller the discretisation constant, the more precise the approximation is. This ensures that the discretisation is sound, meaning that by taking smaller and smaller discretisation constants, the error vanishes.

Corollary 6.19.1 (Soundness). *For a NRC MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ it holds that $\lim_{\delta\to 0} \underline{\overline{\mathbf{R}}}^{\delta} = \underline{\overline{\mathbf{R}}}$.*

Proof. First note that $\lim_{\delta\to 0} \mathfrak{E}^{\mathrm{c}}(k,\delta) = \mathfrak{E}^{\mathrm{f}}(k,\delta) = 0$ for every $k \in \mathbb{N}$. The result then follows by applying the squeeze theorem [Bin82, Thm. 4.10] when taking the limit of Eq. (6.28) as δ approaches zero. □

The soundness of the discretisation technique suggests a sound algorithm for estimating the optimal ETR. The algorithm is based on the computation of the optimal ETR via its characterisation proposed by Thm. 6.9 and later simplified in Lem. 6.10 for Markovian states. Following the characterisation, the algorithm offers to compute the approximate optimal ETR in an iterative manner. In the next section we will see how to run the iterations of the algorithm in more details. Moreover, we provide a complexity analysis for it.

6.3 Algorithm and complexity analysis

This section develops a stable numerical algorithm for computing the optimal ETR that harvests the theoretical results established so far. Furthermore, the complexity analysis of the algorithm is discussed. The algorithm is essentially based on the discretisation technique discussed in Sec. 6.2 as explained next.

6.3.1 An algorithm for computing the optimal ETR

We assume that MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ with a time bound $b \geq 0$ is given. The MRA might be NRC or NRD. We aim to approximately estimate $\underline{\overline{\mathbf{R}}}(v, b)$ for every $v \in V$, in case $\mathcal{R}$ is NRC, and otherwise to recognise it as NRD. The approximation quality is imposed by a given accuracy level enforcing the error to be at most $\epsilon > 0$. As mentioned before, the algorithm computes the discretised optimal ETR, principally via its characterisation as a fixed point (Thm. 6.9) using an iterative method. The algorithm shall first identify whether $\mathcal{R}$ is NRC or NRD. In the former case, it estimates the optimal ETR as $\underline{\overline{\mathbf{R}}}^{\delta}(v, b)$ where δ is chosen such that the given accuracy level is respected. It is thus a key step of the algorithm to find a discretisation constant ensuring the accuracy level.

The starting point to look for an appropriate discretisation constant is provided by Thm. 6.19. Given an accuracy level $\epsilon > 0$, by Eq. (6.28) and (6.29) it is enough to find $\delta > 0$ such that

$$\mathfrak{E}^{\mathrm{c}}(\tfrac{b}{\delta}, \delta) + \mathfrak{E}^{\mathrm{f}}(\tfrac{b}{\delta}, \delta) \leq \epsilon \tag{6.30}$$

It is not hard to see that the left hand side of Eq. (6.30) is increasing in δ. Hence, the tighter the inequality, the larger the discretisation. Making the inequality (6.30) tight leads to an algebraic equation of variable δ. Due to the existence of exponential terms there is no closed form solutions for the equation. To be able to conduct our complexity analysis we need, however to have at least a safe – if not the largest – discretisation constant that respects the accuracy level and can be expressed in terms of model parameters. To this end, we compute safe upper bounds for $\mathfrak{E}^{\mathrm{f}}$ and $\mathfrak{E}^{\mathrm{c}}$ via a linear approximation. We make use of

$$1 + x \geq (1 - \frac{x^2}{2})\,\mathrm{e}^x \tag{6.31}$$

for $x \geq 0$ and $k \in \mathbb{N}$. To prove the inequality we can apply Taylor's theorem to $g(x) = \mathrm{e}^x$. It accordingly holds that

$$\mathrm{e}^x = 1 + x + \frac{x^2}{2}\mathrm{e}^{\eta}$$

for some $\eta \in (0, x)$. The inequality then follows by $\mathrm{e}^x \geq \mathrm{e}^{\eta}$. Using the inequality we give a safe upper bound, first to $\mathfrak{E}^{\mathrm{f}}$.

$$\begin{aligned}
\mathfrak{E}^{\mathrm{f}}(k,\delta) &= \bar{\rho}_{\mathrm{f}} \cdot \Big(1-\mathrm{e}^{-k\delta\bar{\lambda}}\big(1+\bar{\lambda}\delta\big)^{k}\Big) \\
&\leq \bar{\rho}_{\mathrm{f}} \cdot \Big(1-\mathrm{e}^{-k\delta\bar{\lambda}}\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{k}\mathrm{e}^{k\delta\bar{\lambda}}\Big) && \big[* \text{ Eq. (6.31) } *\big] \\
&= \bar{\rho}_{\mathrm{f}} \cdot \Big(1-\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{k}\Big) \\
&\leq \bar{\rho}_{\mathrm{f}} \frac{k\bar{\lambda}^2\delta^2}{2}
\end{aligned}$$

where the last derivation follows from Bernoulli's inequality, i. e. $(1+x)^k \geq 1+kx$, for $x \geq -1$ and $k \in \mathbb{N}$. Since in practice usually a high accuracy level is required, it is safe to assume $\bar{\lambda}\delta \lll 1$. This therefore justifies the usage of Bernoulli's inequality. Finally, put $k = {}^{b}/_{\delta}$ in the above inequality to get

$$\mathfrak{E}^{\mathrm{f}}(\tfrac{b}{\delta},\delta) \leq \frac{\bar{\rho}_{\mathrm{f}}\bar{\lambda}^2 b}{2}\delta \tag{6.32}$$

We use Bernoulli's inequality to provide an upper bound for $\mathfrak{E}^{\mathrm{c}}$ as well.

$$\begin{aligned}
\mathfrak{E}^{\mathrm{c}}(k,\delta) &= k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-i\delta\bar{\lambda}}\big(1+\bar{\lambda}\delta\big)^{i} \\
&\leq k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=0}^{k-1}\mathrm{e}^{-i\delta\bar{\lambda}}\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{i}\mathrm{e}^{i\delta\bar{\lambda}} && \big[* \text{ Eq. (6.31) } *\big] \\
&= k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\sum_{i=0}^{k-1}\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{i} \\
&= k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\frac{1-\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{k}}{1-\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)} \\
&= k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\frac{1-\big(1-{}^{\bar{\lambda}^2\delta^2}/_{2}\big)^{k}}{{}^{\bar{\lambda}^2\delta^2}/_{2}} && (6.33)
\end{aligned}$$

At this point we make use of $(1-x)^k \leq 1-kx+k(k-1)\frac{x^2}{2}$ for $x \in [0,1]$ and $k \geq 1$. This inequality can be proven by Taylor's theorem as follows. It holds by the theorem for $k \geq 2$ that

$$(1-x)^k = 1-kx+\tfrac{kx^2}{2}(k-1)(1-\eta)^{k-2}$$

for some $\eta \in (0,1)$. The inequality is established by taking into account that $(1-\eta)^{k-2} \leq 1$. It is easy to see that the inequality holds also for $k = 0,1$.

Considering that $\bar{\lambda}\delta \ll 1$ we apply the inequality to Eq. (6.33) to obtain

$$\begin{aligned}
\mathfrak{E}^{\mathrm{c}}(k,\delta) &\leq k\delta L^{\mathrm{c}} - \frac{L^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\frac{1-\Big(1-k^{\bar{\lambda}^2\delta^2}/_2+k(k-1)^{\bar{\lambda}^4\delta^4}/_8\Big)}{\bar{\lambda}^2\delta^2/2}\\
&= k\delta L^{\mathrm{c}} - \frac{kL^{\mathrm{c}}}{\bar{\lambda}}\Big(1-\mathrm{e}^{-\bar{\lambda}\delta}\Big)\Big(1-(k-1)\tfrac{\bar{\lambda}^2\delta^2}{4}\Big)\\
&\overset{(\dagger)}{\leq} k\delta L^{\mathrm{c}} - \frac{kL^{\mathrm{c}}}{\bar{\lambda}}\Big(\bar{\lambda}\delta-\tfrac{\bar{\lambda}^2\delta^2}{2}\Big)\Big(1-(k-1)\tfrac{\bar{\lambda}^2\delta^2}{4}\Big)\\
&= L^{\mathrm{c}}\frac{k\bar{\lambda}\delta^2}{2}\Big(1+(k-1)\tfrac{\bar{\lambda}\delta}{2}-(k-1)\tfrac{\bar{\lambda}^2\delta^2}{4}\Big)
\end{aligned}$$

where $(\dagger)$ follows from $1-\mathrm{e}^x \geq x-\frac{x^2}{2}$ for $x \geq 0$. By setting $k = {}^{b}/_{\delta}$, it finally gives

$$\begin{aligned}
\mathfrak{E}^{\mathrm{c}}(\tfrac{b}{\delta},\delta) &\leq L^{\mathrm{c}}b\frac{\bar{\lambda}\delta}{2}\Big(1+(\tfrac{b}{\delta}-1)\tfrac{\bar{\lambda}\delta}{2}-(\tfrac{b}{\delta}-1)\tfrac{\bar{\lambda}^2\delta^2}{4}\Big)\\
&= L^{\mathrm{c}}b\frac{\bar{\lambda}\delta}{2}\Big(1+\tfrac{\bar{\lambda}b}{2}-\tfrac{\bar{\lambda}\delta}{2}(1+\tfrac{\bar{\lambda}b}{2})+\tfrac{\bar{\lambda}^2\delta^2}{4}\Big)\\
&\leq L^{\mathrm{c}}b\frac{\bar{\lambda}\delta}{2}\Big(1+\tfrac{\bar{\lambda}b}{2}-\tfrac{\bar{\lambda}\delta}{2}(1+\tfrac{\bar{\lambda}\delta}{2})+\tfrac{\bar{\lambda}^2\delta^2}{4}\Big) && \Big[*\ \delta \leq b\ *\Big]\\
&= L^{\mathrm{c}}b\frac{\bar{\lambda}\delta}{2}\Big(1+\tfrac{\bar{\lambda}b}{2}-\tfrac{\bar{\lambda}\delta}{2}\Big)\\
&\leq L^{\mathrm{c}}b\frac{\bar{\lambda}\delta}{2}\Big(1+\tfrac{\bar{\lambda}b}{2}\Big) && (6.34)
\end{aligned}$$

Now that we have the upper bounds of $\mathfrak{E}^{\mathrm{f}}$ and $\mathfrak{E}^{\mathrm{c}}$, which are both linear in δ, we can simply take δ such that it respects a given accuracy level ϵ. To this end, we incorporate Eq. (6.32) and (6.34) into Eq. (6.30) to obtain for $\epsilon > 0$

$$\begin{aligned}
&& \epsilon &\geq L^{\mathrm{c}}b\tfrac{\bar{\lambda}\delta}{2}\big(1+\tfrac{\bar{\lambda}b}{2}\big)+\tfrac{\bar{\rho}_{\mathrm{f}}\bar{\lambda}^2 b}{2}\delta\\
\Longleftrightarrow && \delta &\leq \frac{\epsilon}{\frac{\bar{\lambda}b}{2}\big(L^{\mathrm{c}}(1+\frac{\bar{\lambda}b}{2})+\bar{\rho}_{\mathrm{f}}\bar{\lambda}\big)} && (6.35)
\end{aligned}$$

Eq. (6.35) provides an upper bound of on discretisation constant δ so as to assure an accuracy of at least ϵ. Moreover, it provides a lower bound on how many steps must be taken by the discretisation approach to guarantee the accuracy level fixed a priori. This enables us to analyse the numbers of iterations required for the algorithm.

Efficiency. In practice the discretisation constants offered by Eq. (6.35) are usually much smaller than needed. It often happens that the accuracy level is reached even if much larger discretisation constants are taken. It is indeed desirable to find a larger discretisation step provided that it still respects the accuracy level. As a result, a smaller number of iterations is taken and the computation is thereby faster. One solution is to resort to root finding algorithms since, as mentioned before, there is no closed form solution for Eq. (6.30). In our implementation we employ this approach and utilise Newton method to estimate a larger discretisation constant that still respects the accuracy level.

Iterations. We can now estimate the number of iterations to be taken by the discretisation technique. Suppose we have chosen an appropriate δ from a given accuracy level $\epsilon > 0$ using the method discussed above. Hence, it is required to take ${}^{b}/_{\delta}$ iterations to reach the deadline given by the problem. We now explain the computation that is done in each iteration. As discussed before, our algorithm evaluates the discretised optimal ETR via its characterisation as a fixed point. That is, it proceeds according to the following recursion for $k \in \mathbb{N}$:

$$\begin{aligned}\overline{\underline{\mathbf{R}}}^{\delta}(v, k+1) = {} & \left(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\right) \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\, \overline{\underline{\mathbf{R}}}^{\delta}(v', k) \\ & + \left(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\right)\left(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta}\right) + \overline{\underline{\mathbf{R}}}^{\delta}(v, k)\, \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\,\delta} \end{aligned} \tag{6.36}$$

for $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$ and

$$\overline{\underline{\mathbf{R}}}^{\delta}(v, k) = \operatorname*{mix}_{\gamma \in T(v)} \left(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_{\gamma}(v')\, \overline{\underline{\mathbf{R}}}^{\delta}(v', k)\right) \tag{6.37}$$

for $v \in V_{\mathrm{P}}$. The correctness of Eq. (6.36) and (6.37) comes from Lem. 6.10 and Thm. 6.9, respectively. An iteration of the algorithm is comprised of (*i*) computing $\overline{\underline{\mathbf{R}}}^{\delta}(v, k+1)$ for $v \in V_{\mathrm{M}}$ from the previously computed vector $\overline{\underline{\mathbf{R}}}^{\delta}(\cdot, k)$ using Eq. (6.36) and (*ii*) computing the fixed point induced by Eq. (6.36) to obtain $\overline{\underline{\mathbf{R}}}^{\delta}(v, k)$ for $v \in V_{\mathrm{P}}$. The former, i. e. the computation of the objective for Markovian states is performed in a straightforward way via Eq. (6.36). It is referred to as an *M-update*. However, the crucial part of each iteration lies in the latter. This part of the iterations is called a *P-update*. The details of each comes later.

M- and P-updates. The algorithm runs the iterations as follows. At the first iteration ($k = 0$), $\overline{\underline{\mathbf{R}}}^{\delta}(v, 0)$ is set to $\rho_{\mathrm{f}}(\check{\gamma})$ for $v \in V_{\mathrm{M}}$ with $T(v) = \{\check{\gamma}\}$. Therefore an M-update is not relevant here. To conduct the P-update, special attentions should be paid: We first observe by Eq. (6.37) that P-updates only deal with probabilistic states. Hence, they target only the underlying MDP structure of the model, which is induced by probabilistic states. Secondly, in a P-update, we can treat each Markovian state as a deadlock state with final reward of $\overline{\underline{\mathbf{R}}}^{\delta}(v, k)$ since its value stays constant during this phase. All in all, a P-update is nothing but computing the optimal expected total reward in the underlying MDP part of the model with Markovian states playing the role of deadlock states. Formally speaking, the underlying MDP of MA $\mathcal{M} = (V, v_0, T)$ is indeed a new MA obtained by removing its Markovian transitions, i. e. μMDP($\mathcal{M}$) $:= (V, v_0, T_{\mathrm{P}})$. In summary, a P-update is the computation of the fixed point suggested by Eq. (6.37) on μMDP($\mathcal{M}$). There are different approaches to conduct a P-update, it is for instance possible to employ reward optimisation algorithms working on MDPs. However, we take a different view to solve the problem, namely by employing *goal-bounded analysis* while the Markovian states are taken as the goal set. In this view, we want to optimise reward acquisition until reaching some state in the goal set. This is referred to as the *optimal expected goal-bounded reward*, solved in [Guc+14b]. We encode the P-update problem as this problem in the next lemma.

Lemma 6.20. *For a given MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$ and discretisation constant $\delta > 0$, the P-update at iteration k can be done via computing the* optimal expected goal-bounded rewards *of* μMDP($\mathcal{M}$) *under reward structure $\rho^{(k)} = (\rho_{\mathrm{t}}, \rho_{\mathrm{i}}, \rho_{\mathrm{f}}^{(k)})$ with $\rho_{\mathrm{f}}^{(k)}(v) = \overline{\underline{\mathbf{R}}}^{\delta}(v,k)$ for $v \in V_{\mathrm{M}}$ and zero otherwise. Moreover, the goal set contains all Markovian states.*

Proof. The proof is simply by comparing the fixed point characterisations of the two objectives. The fixed point computed in order to carry out the P-update is given by Eq. (6.37). The other one can be obtained via a slight extension of the fixed point given in [Guc+14b, Thm. 1], i. e. letting goal states have arbitrary positive final rewards, instead of zero. It is not hard to see thus both fixed point characterisations are equivalent and thereby their least fixed points are the same. This completes the proof. □

Note also that in the above lemma we assign the final rewards to Markovian states instead of Markovian transitions, simply because the latter do not exist in the underlying MDP. To make a clear view of what is going to be computed in a P-update, we reformulate Eq. (6.37) as a Bellman equation. We first assume an arbitrary total ordering of the states of the MRA that allows us to consider any function $g \colon V \to \overline{\mathbb{R}}$ as vector $\boldsymbol{g} \in \overline{\mathbb{R}}^{|V|}$ with $\boldsymbol{g}_v = g(v)$ for every $v \in V$. We keep this ordering later when we discuss the algorithm. A P-update is then carried out by computing the *least fixed point* of the *Bellman optimality operator* $\bar{\mathcal{T}}^f : \overline{\mathbb{R}}_{\geq 0}^{|V|} \to \overline{\mathbb{R}}_{\geq 0}^{|V|}$ such that for every vector $\boldsymbol{g} \in \overline{\mathbb{R}}_{\geq 0}^{|V|}$:

$$\bar{\mathcal{T}}^f(\boldsymbol{g}_v) = \begin{cases} \operatorname{mix}_{\gamma \in T(v)} \left(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathrm{d}_\gamma(v') \cdot \boldsymbol{g}_{v'}\right) & v \in V_{\mathrm{P}} \\ \boldsymbol{f}_v & v \in V_{\mathrm{M}} \end{cases} \tag{6.38}$$

where $\boldsymbol{f} \in \overline{\mathbb{R}}_{\geq 0}^{|V|}$ is the vector of final rewards for Markovian states. We can finally conclude that the P-update at iteration k is the least fixed point of $\bar{\mathcal{T}}^{f^{(k)}}$ with $\boldsymbol{f}_v^{(k)} = \overline{\underline{\mathbf{R}}}^{\delta}(v,k)$ for all $v \in V$.

Computing the maximal null reward. The result of Lem. 6.20 together with Eq. (6.38) not only provides a way to compute a P-update, but also, as a special case, offers a solution for computation of the maximal null reward. It is required to know this value to determine whether the model is NRC or NRD. Moreover, if the model is NRC that value is again needed for computation of L^{c} to be used for searching an appropriate discretisation constant, for instance by Eq. (6.35). Recall that the maximal null reward is the maximum reward that a resource preserving state can earn up to reaching some resource consuming state. In the time-bounded setting, it is translated to the maximum reward starting from probabilistic states up to reaching some Markovian state. As a result the maximal null reward can be obtained by conducting a P-update while the final reward of Markovian states, all are set to zero.

The algorithm. We now integrate all pieces together to obtain the algorithm for computing an approximation of the optimal ETR with a proven error bound. The algorithm, displayed in Alg. 6.1, takes as input an arbitrary MRA, a deadline, an accuracy level and the optimisation goal which might either be "min" or "max". It then determines whether the MRA is NRC or NRD by computing the maximal null reward. In the former case, it estimates the optimal ETR within the given deadline while respecting the accuracy level. As mentioned before, we take an arbitrary total ordering into account in order to do all of the computations on vectors.

Alg. 6.1 first computes the parameters of the MRA in lines 2 and 3. In particular, it evaluates the maximal null reward to determine whether it is finite or infinite. In the latter case the model is NRD and therefore the algorithm terminates. Otherwise, the algorithm searches for an appropriate discretisation constant that respects the given accuracy level for both absorbing and non-absorbing models. This can be easily done using the linear approximations given by Eq. (6.35). However, a better solution is to employ a numerical root finder like Newton algorithm to approximately find $\delta > 0$ that makes inequality (6.30) tight. In practice, this method is favourable since it leads to larger discretisation constants and thereby a smaller number of iterations. The computed discretisation constant should be adjusted in a way that b is reached after taking an integer number of steps of length δ, specified in the algorithm by K. This adjustment must indeed respect the accuracy level. Thereafter, the iterations of the algorithm are carried out. Each iteration contains an M-update followed by a P-update, computed by functions `Mupdate` and `Pupdate`, respectively. The result of each iteration is stored in vector $\boldsymbol{x}$. After the last iteration, the vector contains the ϵ-approximate optimal ETR enjoying the given accuracy level as described in the output of the algorithm.

Fixed point computation. The crucial part of each iteration is to perform the P-update, which is done by function `Pupdate` in the algorithm. It computes the

Algorithm 6.1: Computing ϵ-approximate optimal ETR

```
Input: MRA $\mathcal{R} := (\mathcal{M}, \mathbb{1}_{T_{\mathrm{M}}}, \rho)$, horizon $b \geq 0$, accuracy level $\epsilon > 0$,
       optimisation goal (min or max)
Output: Report if $\mathcal{R}$ is NRD, otherwise vector $\{\boldsymbol{x}_v\}_{v \in V}$ such that
        $\left|\boldsymbol{x}_v - \overline{\underline{\mathbf{R}}}(v, b)\right| \leq \epsilon$, or if $\mathcal{R}$ is absorbing $0 \leq \overline{\underline{\mathbf{R}}}(v, b) - \boldsymbol{x}_v \leq \epsilon$
begin
    $\bar{\rho}_{\mathrm{t}} = \max_{\gamma \in T_{\mathrm{M}}} \rho_{\mathrm{t}}(\gamma)$, $\bar{\rho}_{\mathrm{i}} = \max_{\gamma \in T_{\mathrm{M}}} \rho_{\mathrm{i}}(\gamma)$, $\bar{\rho}_{\mathrm{f}} = \max_{\gamma \in T_{\mathrm{M}}} \rho_{\mathrm{f}}(\gamma)$, $\bar{\lambda} = \max_{\gamma \in T_{\mathrm{M}}} \mathrm{r}_\gamma$
    $\boldsymbol{x}$ = Pupdate($\mathbf{0}$), $r_0 = \max_{v \in V} \boldsymbol{x}_v$        // the maximal null reward
    if $r_0 = \infty$ then
        print "The model is NRD"
        exit
    end
    $L^{\mathrm{c}} = \bar{\rho}_{\mathrm{t}} + \bar{\lambda} \cdot (\bar{\rho}_{\mathrm{i}} + r_0)$
    choose some $\delta > 0$ that
        • respects Eq. (6.30) and
        • there exists $K \in \mathbb{N}$ such that $b = K\delta$
    foreach $v \in V_{\mathrm{M}}$ do $\boldsymbol{x}_v = \rho_{\mathrm{f}}(\check{\gamma})$            /* Initialisation */
    $\boldsymbol{x}$ =Pupdate ($\boldsymbol{x}$)
    for $i = 1$ to $K$ do                                 /* Iterations */
        $\boldsymbol{x}$ =Mupdate ($\boldsymbol{x}$)
        $\boldsymbol{x}$ =Pupdate ($\boldsymbol{x}$)
    end
end
Function Mupdate($\boldsymbol{x}$):
    foreach $v \in V$ do $\boldsymbol{x}'_v = 0$
    foreach $v \in V_{\mathrm{M}}$ do
        $\boldsymbol{x}'_v = \big(1 - \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}\big)\big(\sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v')\boldsymbol{x}_{v'} + \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\big) + \boldsymbol{x}_v \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\delta}$
    return $\boldsymbol{x}'$
end
Function Pupdate($\boldsymbol{x}$):
    return the least fixed point of $\bar{\underline{\mathcal{T}}}^{\boldsymbol{x}}$, given in Eq. (6.38)
end
```

least fixed point of the Bellman optimality operator given in Eq. (6.38). There are different ways to compute the fixed point, which will be discussed in detail in Sec. 7.1.2.

Soundness. We explain at this point why Alg. 6.1 is sound. We first stress the point that the algorithm computes the discretised optimal ETR via its characterisation given in Thm. 6.9 and Lem. 6.10. Secondly, the selection of discretisation constant δ on line 9 of the algorithm assures that the discretisation error is at most ϵ. This is justified by Thm. 6.19. As a result what the algorithm computes, as claimed, is in the ϵ-neighbourhood of the optimal ETR for every state of the input MRA. The result is concluded in the next theorem.

Theorem 6.21. *Alg. 6.1 is* sound.

Proof. The claim follows from Thm. 6.9, Lem. 6.10 and Thm. 6.19. □

Special cases. As mentioned before, various kinds of analyses can be encoded as the optimal ETR. However, Alg. 6.1 conduct them in the most general case. It is possible to adapt the algorithm for special classes of analyses and models. This may lead to a more efficient analysis. A notable example is computing reachability and cumulative reward for an absorbing MRA. As it can be implied from the output of the algorithm, the computation is more precise for absorbing models. It follows from Thm. 6.19 that the optimal ETR is always greater than or equal to its approximation for absorbing models whereas that is not necessarily the case for non-absorbing models. This might for instance make a decision procedure induced by a logical formula more efficient for absorbing models. It is particularly useful when solving the problem of the *optimal expected time-bounded cumulative reward* and the *optimal time-bounded reachability*. For the former Def. 6.17 implies that the model is absorbing as the final rewards of transitions are all zero. For the latter we apply the technique of *making goal states absorbing* to obtain an absorbing model.

Here we discuss briefly how the technique of *making goal states absorbing* [Bai+03] is employed for computing time-bounded reachability. Recall that in time-bounded reachability the objective is to compute the optimal probability to reach some state in a given goal set within a deadline. It is encoded in our setting by making all states in the goal set absorbing, i. e. removing all of their outgoing transitions and replacing them with a Markovian self loop with an arbitrary rate and with final reward of one. It is known (cf. [Neu10, Thm. 6.7]) that this encoding preserves time-bounded reachability of IMCs. The extension to MRAs is straightforward. Finally, Alg. 6.1 can be applied after making the model absorbing. The error bound given by Eq. (6.29) in Thm. 6.19 thus holds in this setting.

Extension. We have recently extended Alg. 6.1 so as to compute the same measure in two-player stochastic games [Bra+14; Bra+15]. We have then exploited the extension in an abstraction refinement framework for MRAs. There, an MRA is abstracted into a smaller two-player game such that analysing the game allows to deduce safe bounds on the optimal ETR of the original model.

6.3.2 Complexity analysis

In this section we provide a complexity analysis of Alg. 6.1. In some parts of the algorithm there are some levels of freedom to choose variant methods for performing specific tasks. Precisely speaking, to be able to analyse the complexity we need to make it clear how we compute the discretisation constant on line 9 and also the P-updates in `Pupdate` function. To this end we assume Eq. (6.35) is used for computing the discretisation constant and *linear programming* is employed to conduct P-updates. Note that these assumptions are not the best choices in practice. For instance we known that applying Newton steps in Eq. (6.30) almost always gives a larger discretisation step size than the one offered by Eq. (6.35). This leads to a smaller number of iterations. Moreover, *value iteration* is know [For+11] to be empirically much faster than solving *linear programming* for a P-update. However, these choices not only simplify the complexity analysis, but also provide a pessimistic hardness for Alg. 6.1 which in fact is still polynomial in the size of the model.

The first step of our complexity analysis is to estimate the number of iterations that the algorithm takes. Assume a fixed accuracy level $\epsilon > 0$, then from Eq. (6.35) the following discretisation constant respects the accuracy level.

$$\delta' = \frac{\epsilon}{\frac{\bar{\lambda}b}{2}\left(L^{\mathrm{c}}(1+\frac{\bar{\lambda}b}{2})+\bar{\rho}_{\mathrm{f}}\bar{\lambda}\right)} \tag{6.39}$$

With this, $\lceil b/\delta' \rceil$ iteration(s) need to be taken to cross the time bound, which equals

$$\left\lceil \frac{\bar{\lambda}b^2}{2\epsilon}\left(L^{\mathrm{c}}(1+\frac{\bar{\lambda}b}{2})+\bar{\rho}_{\mathrm{f}}\bar{\lambda}\right)\right\rceil \tag{6.40}$$

In each iteration an M-update and a P-update is carried out. For each Markovian state in an M-update $\mathcal{O}(|V|)$ operations are executed since the Markovian state might be connected to all other states. The complexity of an M-update is thus $\mathcal{O}(|V_{\mathrm{M}}||V|)$. It remains to analyse the complexity of a P-update. To formulate P-update as a linear programming [BK08; Put05] we introduce a variable for each probabilistic state and impose a constraint for each probabilistic transition. Let $\mathcal{LP}(n,m)$ be the complexity of solving linear programming with n variables and m constraints, then $\mathcal{LP}(|V_{\mathrm{P}}|,|T_{\mathrm{P}}|)$ is the complexity of conducting P-update via solving the corresponding linear programming. Note also that the complexity of solving linear programming is known [Kha80] to be polynomial in the number of variables and constraints. Finally we can obtain the complexity of Alg. 6.1, which is

$$\mathcal{O}\left(\frac{\bar{\lambda}b^2}{2\epsilon}\left(L^{\mathrm{c}}(1+\frac{\bar{\lambda}b}{2})+\bar{\rho}_{\mathrm{f}}\bar{\lambda}\right)\cdot\left(|V_{\mathrm{M}}||V|+\mathcal{LP}(|V_{\mathrm{P}}|,|T_{\mathrm{P}}|)\right)\right) \tag{6.41}$$

Discussion. The complexity of Alg. 6.1 as shown above is polynomial in the parameters of the input MRA, provided that the accuracy level is fixed. However the algorithm may not be efficient in practice. In particular the number of iterations is quadratic in $\bar{\lambda}$ and b. This implies that the number of iterations can grow very fast when increasing the deadline or the largest exit rate of the model.

One solution to this problem is to change the assumption made by the discretisation in order to achieve the same accuracy level but with much less number of iterations. It has been shown [HH13a; HH15] for IMCs that the number of iterations drastically reduces when we let more Markovian jumps lie in a discretisation step. In the second-order discretisation, which allows at most two jumps in a discretisation step, the number of iterations is in the order of $\sqrt{(\bar{\lambda}b)^3}$. This method, on the other hand, increases the per iteration complexity since in this setting the decision of the (ϵ-)optimal strategy may change between the first and the second Markovian jumps. The point at which the decision is changed, named *swapping point*, can be computed by a root finding algorithm. The complexity of the second-order approximation that utilises the bisection method as the root finding algorithm is discussed in [HH15]. In practice also, the second-order approximation is shown to be empirically much faster than the first-order approximation, which sticks to the same principles as the ones in Alg. 6.1. However, adapting this method to MRA seems not to be easy. The main complication arises when each decision that can be made by probabilistic states must be considered to estimate swapping points. The number of decisions to be analysed for computing the swapping point in general may be exponential in the size of the model. This may not happen for IMCs since their interactive part has much simpler structure, namely it only contains transitions with single successors. This makes the number of decisions to be considered for swapping point computation at most as large as the number of states in the model. Nevertheless for the MRAs with relatively small number of transitions, the second-order approximation works well, especially because the number of decisions to be considered is small.

Another solution is to use the idea of adaptive discretisation, where the discretisation constant is adapted during the computation according to error estimation. With this we achieve the predetermined accuracy level while trying to minimise the computational steps. The method is known to be effective for the integration of ordinary differential equations, for instance when Runge-Kutta is used [Pre+07, Sec. 17.2]. It is also employed for the computation of the optimal ETR of CTMDPs for the class of late strategies [Buc+11]. This method is shown to be highly effective among various industrial and academic case studies [But+15]. We strongly believe that adaptive discretisation with proved error bounds can be developed in our setting. It is however left for future work.

6.4 From time to resource

We have seen thus far how to compute the optimal ETR for an MRA. As we mentioned before, it was introduced not only because it addresses the computation of a useful class of analyses, but also because it provides the basis for the computation of a more general class, namely the optimal ERR. In this section we therefore bring resources into play and discuss how to compute the optimal ERR using the already known solution for the optimal ETR.

The discretisation approach described in Sec. 6.2 cannot be easily lifted to the resource-bounded setting. The approach to compute the optimal ETR is through discretisation of its fixed point characterisation. That is to say, the Volterra integral equation (Eq. (6.1a)) is discretised over the time horizon from 0 to b. By comparing the structure of the fixed point described in Thm. 6.2 with the one in Thm. 5.2 it turns out that they do not look particularly different. However, the discretisation cannot be efficiently adapted to the resource-bounded setting. The main reason is that resources might be spent with different rates among different states whereas the time passage is globally the same for all (Markovian) states. It helps in particular to designate the same discretisation scheme among all states, which is not easily possible in the resource-bounded setting.

We propose a transformation from the time- to the resource-bounded setting. During the transformation both the model and its resource and reward structures will be changed. The transformation reduces the problem of computing the optimal ERR in the original model into the problem of computing the optimal ETR in the transformed model. In other words, we transform an MRA into another one such that the optimal ERR of the original model coincides with the optimal ETR of the transformed one. We can then apply Alg. 6.1 proposed in Sec. 6.3 to compute the optimal ETR of the transformed model.

6.4.1 The transformation

The core idea of the transformation comes from Beaudry [Bea78]. It exploits rate scaling as the main technique for *resource-to-time* transformation in CTMCs. More precisely, it scales transition rates reciprocal to their resource consumption rates. In this way, the time passage of the transitions is stretched or compressed according to their resource consumption. Therefore, for a high resource consuming transition time passes slowly so as to simulate a high resource consumption with a long sojourn time. The intuition for a low resource consuming transition is similar. At the end, the rate scaling enables us to simulate resource consumption with time passage. In other words, the resource consumption in the original model coincides with the time passage in the transformed model. This idea was later extended to CTMDPs under the class of time-abstract strategies [Bai+08]. The technique is however restricted to *uniform* CTMDPs, which are CTMDPs in which all states have the same exit rate.

Requirements. We aim to generalise the resource-to-time transformation of Beaudry to MRAs for computing the optimal ERR. In our setting, we want to lift the restriction of [Bai+08] on the class of strategies and on the uniformity of the model. Moreover, we want our transformation to support states with zero resource consumption as it is natural and useful to have such states in a system. We shall take into account that the transformation may manipulate the transient reward gained by a state by changing the distribution of sojourn time. Altogether, we address a resource-to-time transformation that

1. supports general class of MRAs, in particular containing resource preserving states and transitions,

2. preserves total reward collection under the class of generic measurable strategies.

The idea. To fulfil the above requirements we make use of the specific structure of MRAs. In comparison to CTMCs and CTMDPs, MRAs may contain probabilistic states, states that fire their transitions instantaneously. We exploit it to meet the first requirement; more precisely to deal effectively with resource preserving transitions. Such transitions with zero resource consumption suffer from having infinite rate after the rate scaling. This in fact justifies turning them into probabilistic states since infinite rate implies infinitely fast execution.

The transformation. We follow the core idea of the rate scaling, namely we scale the rate of resource consuming transitions reciprocal to their resource consumption rate. In order to enable the transformation to preserve transient rewards, we need to scale transient reward rates in the same way. It remains to explain how to tackle resource preserving transitions. According to our intuition, such transitions must be infinitely fast, leaving no chance for consuming any resource. As mentioned before, this can be indeed imitated by probabilistic transitions. But in this approach, transient rewards need to be adapted accordingly. Since probabilistic transitions cannot gain any transient reward, the transient reward obtained by the original resource preserving transition needs to be properly transferred to its corresponding probabilistic transition. To do this, we add the expected transient reward gained by the original transition to the instantaneous reward of the transformed transition. Altogether, the formal description of the transformation is given in the next definition.

Definition 6.22 (Resource-to-time transformation). *Let $\mathcal{R} := (\mathcal{M}, \varrho, \rho)$ be an MRA. The* (resource-to-time) transformed MRA *is defined as $\mathcal{R}^{\varrho} = (\mathcal{M}^{\varrho}, \mathbb{1}_{T_{\mathrm{M}}}, \rho^{\varrho})$ with $\mathcal{M}^{\varrho} := (V, v_0, T^{\varrho})$ and $\rho^{\varrho} := (\rho_{\mathrm{t}}^{\varrho}, \rho_{\mathrm{i}}^{\varrho}, \rho_{\mathrm{f}})$ such that*

$$\begin{aligned} T^{\varrho} &= T_{\mathrm{P}} \uplus \left\{ (\mathsf{v}_{\gamma}, \natural, \mathsf{d}_{\gamma}) \,|\, \gamma \in T_{\mathrm{M}} \cap T_{\mathrm{rp}} \right\} \\ &\qquad \uplus \left\{ (\mathsf{v}_{\gamma}, \tfrac{\mathsf{r}_{\gamma}}{\varrho(\gamma)}, \mathsf{d}_{\gamma}) \,|\, \gamma \in T_{\mathrm{M}} \cap T_{\mathrm{rc}} \right\} \\ \rho_{\mathrm{t}}^{\varrho}(\gamma) &= \begin{cases} \frac{\rho_{\mathrm{t}}(\gamma)}{\varrho(\gamma)} & \gamma \in T_{\mathrm{rc}} \\ 0 & \textit{otherwise} \end{cases} \end{aligned}$$

$$\rho_{\mathrm{i}}^{\varrho}(\gamma) = \begin{cases} \rho_{\mathrm{i}}(\gamma) + \frac{\rho_{\mathrm{t}}(\gamma)}{\mathrm{r}_\gamma} & \gamma \in T_{\mathrm{M}} \cap T_{\mathrm{rp}} \\ \rho_{\mathrm{i}}(\gamma) & \textit{otherwise.} \end{cases}$$

We can summarise the transformation as follows. A resource consuming transition in the original model turns into a Markovian transition in the transformed model with its rate and transient reward scaled inverse proportional to its resource consumption. A Markovian resource preserving transition becomes a probabilistic transition with the instantaneous reward increased by the expected transient reward of the original transition. This can be computed using the probability density function of the exponential distribution, which leads to $\frac{\rho_{\mathrm{t}}(\gamma)}{\mathrm{r}_\gamma}$ for transition $\gamma \in T_{\mathrm{rp}} \cap T_{\mathrm{M}}$. Probabilistic transitions stay untouched under the transformation. We can at the end conclude that $T_{\mathrm{M}}^{\varrho} = T_{\mathrm{rc}}$ and $T_{\mathrm{P}}^{\varrho} = T_{\mathrm{rp}}$.

MRAs are obviously closed under the transformation. Additionally, their structure and their size, i. e. the number of states and transitions, stay unchanged across the transformation. More importantly, there is a one-to-one correspondence between the states and transitions of the original and the transformed model. Therefore, the transformation is linear in the size of input model and moreover, it is efficiently computable.

6.4.2 Measure preservation

The resource-to-time transformation exhibits certain features. Apart from the closure and the efficiency discussed in the previous section, it features measure preservation. Measure preservation enables it to be used for computing the optimal ERR. More precisely, the transformation reduces the problem of computing the optimal ERR to computing the optimal ETR. The result is depicted in the next theorem.

Theorem 6.23 (Measure preservation)**.** *Let $\mathcal{R}$, ϱ and $\mathcal{R}^{\varrho}$ be as defined in Def. 6.22, then for all $v \in V$ and every $b \geq 0$ it holds that*

$$\overline{\underline{\mathbf{R}}}_{\mathcal{R}}(v,b) = \overline{\underline{\mathbf{R}}}_{\mathcal{R}^{\varrho}}(v,b)$$

Proof. The proof is done by showing that the original and the transformed models have indeed the same fixed point characterisation for the optimal ERR and ETR, respectively. We start with the fixed point characterisation of the optimal ERR for the original MRA, described in Thm. 5.2 for different cases. We first look at $v \in T_{\mathrm{rc}}$ with $T(v) = \{\check{\gamma}\}$ and use Eq. (5.20a) to write

$$\begin{aligned} \overline{\underline{\Omega}}(F)(v,b) = & \int_0^{b/\varrho(\check{\gamma})} \mathrm{r}_{\check{\gamma}} \cdot \mathrm{e}^{-\mathrm{r}_{\check{\gamma}}\, t} \sum_{v' \in V} \mathrm{d}_{\check{\gamma}}(v') \cdot F\big(v', b - \varrho(\check{\gamma}) \cdot t\big)\, \mathrm{d}t \\ & + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathrm{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma})\Big)\Big(1 - \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}}\Big) + \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathrm{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \end{aligned}$$

We then apply a variable substitution according to the transformation given in Def. 6.22. Let $\check{\gamma}^{\varrho} \in T^{\varrho}$ be the transition in the transformed model that is corresponding to $\check{\gamma}$, then by Def. 6.22 their rates and transient rewards are related by $\mathsf{r}_{\check{\gamma}^{\varrho}} = {}^{\mathsf{r}_{\check{\gamma}}}/_{\varrho(\check{\gamma})}$ and $\rho_{\mathrm{t}}(\check{\gamma}^{\varrho}) = {}^{\rho_{\mathrm{t}}(\check{\gamma})}/_{\varrho(\check{\gamma})}$. Other elements of the transitions are the same i. e. $\mathsf{v}_{\check{\gamma}^{\varrho}} = \mathsf{v}_{\check{\gamma}}$, $\mathsf{d}_{\check{\gamma}^{\varrho}} = \mathsf{d}_{\check{\gamma}}$, $\rho_{\mathrm{i}}^{\varrho}(\check{\gamma}^{\varrho}) = \rho_{\mathrm{i}}(\check{\gamma})$ and $\rho_{\mathrm{f}}(\check{\gamma}^{\varrho}) = \rho_{\mathrm{f}}(\check{\gamma})$. We then introduce variable τ with $\tau := \varrho(\check{\gamma})\, t$. Taking $\mathrm{d}\tau = \varrho(\check{\gamma})\, \mathrm{d}t$ into account we express the above equation in terms of τ:

$$
\begin{aligned}
\overline{\underline{\Omega}}(F)(v,b) = & \int_0^b \mathsf{r}_{\check{\gamma}} \cdot \mathrm{e}^{-{}^{\mathsf{r}_{\check{\gamma}}}/_{\varrho(\check{\gamma})}\tau} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, F(v', b-\tau) \tfrac{1}{\varrho(\check{\gamma})}\, \mathrm{d}\tau \\
& + \Big(\frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) \Big) \Big(1 - \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \Big) + \rho_{\mathrm{f}}(\check{\gamma})\, \mathrm{e}^{-\frac{\mathsf{r}_{\check{\gamma}} \cdot b}{\varrho(\check{\gamma})}} \\
= & \int_0^b \mathsf{r}_{\check{\gamma}^{\varrho}} \cdot \mathrm{e}^{-\mathsf{r}_{\check{\gamma}^{\varrho}}\, \tau} \sum_{v' \in V} \mathsf{d}_{\check{\gamma}^{\varrho}}(v')\, F(v', b-\tau)\, \mathrm{d}\tau \\
& + \Big(\frac{\rho_{\mathrm{t}}^{\varrho}(\check{\gamma}^{\varrho})}{\mathsf{r}_{\check{\gamma}^{\varrho}}} + \rho_{\mathrm{i}}^{\varrho}(\check{\gamma}) \Big) \Big(1 - \mathrm{e}^{-\mathsf{r}_{\check{\gamma}^{\varrho}}\, b} \Big) + \rho_{\mathrm{f}}(\check{\gamma}^{\varrho})\, \mathrm{e}^{-\mathsf{r}_{\check{\gamma}^{\varrho}}\, b}
\end{aligned} \tag{6.42}
$$

It can be already seen that the above equation looks exactly like Eq. (6.1a) in Thm. 6.2. In order to complete the proof we still need to consider $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$ and combine it with the case $v \in V_{\mathrm{P}}$. Once more, assume $\check{\gamma}^{\varrho} \in T^{\varrho}$ is the transformed version of $\check{\gamma} \in T(v)$ for $v \in V_{\mathrm{rp}} \cap V_{\mathrm{M}}$. Following from Def. 6.22, it holds that $\rho_{\mathrm{i}}^{\varrho}(\check{\gamma}^{\varrho}) = \rho_{\mathrm{i}}(\check{\gamma}) + {}^{\rho_{\mathrm{t}}(\check{\gamma})}/_{\mathsf{r}_{\check{\gamma}}}$. Moreover, $\check{\gamma}^{\varrho}$ becomes a probabilistic state of the transformed model with the same source and distribution as $\check{\gamma}$. Therefore, by Eq. (5.20b) we have

$$
\begin{aligned}
\overline{\underline{\Omega}}(F)(v,b) &= \frac{\rho_{\mathrm{t}}(\check{\gamma})}{\mathsf{r}_{\check{\gamma}}} + \rho_{\mathrm{i}}(\check{\gamma}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}}(v')\, F(v', b) \\
&= \rho_{\mathrm{i}}^{\varrho}(\check{\gamma}^{\varrho}) + \sum_{v' \in V} \mathsf{d}_{\check{\gamma}^{\varrho}}(v')\, F(v', b)
\end{aligned} \tag{6.43}
$$

As mentioned before, probabilistic transitions are not changed by the transformation. Thus, we can write for $v \in V_{\mathrm{P}}$ that

$$
\begin{aligned}
\overline{\underline{\Omega}}(F)(v,b) &= \operatorname*{mix}_{\gamma \in T(v)} \Big(\rho_{\mathrm{i}}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v') \cdot F(v', b) \Big) \\
&= \operatorname*{mix}_{\gamma \in T^{\varrho}(v)} \Big(\rho_{\mathrm{i}}^{\varrho}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v') \cdot F(v', b) \Big)
\end{aligned} \tag{6.44}
$$

Since $\check{\gamma}^{\varrho}$ is the only transition of v and now it is probabilistic in the transformed model, we can combine Eq. (6.43) and (6.45) to obtain

$$
\overline{\underline{\Omega}}(F)(v,b) = \operatorname*{mix}_{\gamma \in T^{\varrho}(v)} \Big(\rho_{\mathrm{i}}^{\varrho}(\gamma) + \sum_{v' \in V} \mathsf{d}_{\gamma}(v') \cdot F(v', b) \Big) \tag{6.45}
$$

Now we combine Eq. (6.42) and (6.45) to constitute a new representation of operator $\overline{\underline{\Omega}}$. In this view, $\overline{\underline{\Omega}}$ is identical to the operator given Thm. 6.2. In the one hand, by Thm. 5.2, $\overline{\underline{\mathbf{R}}}_{\mathcal{R}}$ is the least fixed point of $\overline{\underline{\Omega}}$. With the new representation and by Thm. 6.2 on the other hand, $\overline{\underline{\mathbf{R}}}_{\mathcal{R}\varrho}$ is the least fixed point of $\overline{\underline{\Omega}}$. This complete the proof. □

Thm. 6.23 offers a solution for the computation of the optimal ERR. Given an MRA with resource function $\varrho \colon T \to \mathbb{R}_{\geq 0}$ and resource budget $b \geq 0$, we first apply the transformation described by Def. 6.22. The transformation can be efficiently computed in linear time in the size of the MRA. We are left then with the computation of the optimal ETR of the transformed MRA. At this point, we can apply Alg. 6.1 to compute the objective up to an arbitrary accuracy level. The complexity of computing ϵ-optimal ERR in this way coincides, indeed with the complexity of Alg. 6.1 applied to the transformed MRA, as discussed in Sec. 6.3.2.

6.5 Discussion

This chapter addressed the computation of the optimal ERR via a sound discretisation approach together with a measure preserving resource-to-time transformation. Here we remark on the main points discussed in this chapter.

(*i*) We defined the *optimal ETR* (Def. 6.1) as the special case of the optimal ERR when time is the resource.

(*ii*) We proposed a discretisation scheme, introduced the *discretised optimal ETR* as the limit of a sequence of functions by Def. 6.8 and also established its characteristics as a fixed point (Thm. 6.9).

(*iii*) We studied the error induced by the discretisation scheme and established a lower and an upper bound on it (Thm. 6.19).

(*iv*) We developed a sound algorithm (Alg. 6.1) for computing the optimal ETR via the discretisation scheme.

(*v*) We provided a worst-case complexity analysis for the algorithm and showed that it can be implemented to run in polynomial time.

(*vi*) We proposed an efficient resource-to-time transformation that reduces the computation of the optimal ERR to that of the optimal ETR.

Contributions. This chapter made two major contributions to the analysis of MRAs. Firstly, as mentioned in points (*ii*), (*iii*) and (*iv*), it developed a sound discretisation approach with strict error bound that leads to a polynomial time algorithm for computing the optimal ETR. The algorithm can approximate the optimal ETR up to a given precision level. This is a generalisation of our previous works [HH12; Guc+13; Guc+14a; Guc+14b; Bra+15]. Secondly, it proposed an efficient resource-to-time transformation that reduces the problem of computing the optimal ERR to that of computing the optimal ETR. This ingredient was published in [Hat+15]. Combining these two contributions together enables us to compute an ϵ-approximation of the optimal ERR for MRAs in an efficient and sound way. In the sequel, we elaborate on the existing algorithms and techniques related to our contributions.

6.5.1 Existing algorithms for CTMDP

To the best of our knowledge, the work described here is the only work developing a solution for time- and resource-bounded analysis of MRAs. There are, however, a number of related algorithms, especially in the context of CTMDPs. Here we discuss them in more details.

Zhang-Neuhäusser algorithm [ZN10; Neu10] computes the maximal time-bounded reachability of IMCs and CTMDPs for both early and late strategies. Their approach is to discretise the fixed point characterisation of the maximal time-bounded reachability approach. To this end, the time horizon is split into a finite number of chunks. The reachability probabilities are then evaluated at the equidistant points induced by the splitting. The discretisation assumes that at most one Markovian jump can happen within each chunk. This assumption approximates reachability probabilities by a piece-wise exponential function emerging from the discretisation of the fixed point characterisation. This can yield an a priori error bound. Our discretisation can be seen as the extension of theirs for a more general class of analysis, namely the optimal ERR, and a more general class of models, namely MRAs. Furthermore, in [HH13a; HH15] we extended the Zahng-Neuhäusser algorithm for timed reachability analysis of IMCs by letting more Markovian jumps occur in each chunk. We showed by an empirical evaluation that this is superior to the original Zahng-Neuhäusser algorithm.

Buchholz-Schulz algorithm [BS11; Buc+11] employs both discretisation and *uniformisation* [Jen53] for computing the maximal time-bounded reward of a CTMDP under the class late strategies. The core idea behind the approach comes from discretising the differential characterisation of the maximal reward function, given by [Mil68], and solving it using uniformisation. This approach does not support transition dependent rewards. In this approach the discretisation is done in an adaptive way, meaning that the length of chunks are not necessarily the same. In each chunk, an under and an over approximation of the maximal reward is computed via uniformisation. The error bound at each iteration is not clear a priori but only after computing the approximations. Depending on the shape of the current approximations, the length of the next chunk can be prolonged until the error allowed for this step is reached. This greatly reduces the number of iterations, relative to the non-adaptive setting. The successor of the algorithm [BDS17] provides support for transition dependent rewards in addition to an efficiency improvement.

Polynomial approximation algorithm [Fea+11; Fea+16] computes the optimal time-bounded reachability of a CTMDP (and Markov games) under the class of late strategies. It discretises the differential characterisation of the optimal time-bounded reachability using a collocation method similar to the Runge-Kutta algorithm for solving ordinary differential equations. However, in contrast to most collocation methods, it provides an a priori error bound. The reachability probabilities in each chunk are approximated by polynomials. The higher the degree of the polynomials, the higher is the computational effort, but the number of chunks required to assure an a priori error bound drastically decreases. This is the main advantage of using this algorithm.

Fu's algorithm [Fu14a; Fu14b] provides a naïve discretisation approach with a priori error bound for computing the maximal resource-bounded reachability of CTMDPs with multiple resources. This method approximates the objective by a piece-wise linear function. In this regard, it is similar to the first order version of the polynomial approximation algorithm discussed above. It thus requires a high number of iterations to assure a predefined accuracy level. This method has not been implemented in practice.

Baier et al. algorithm [Bai+05] is one of the earliest attempt to attack the problem of timed reachability in CTMDPs. The algorithm is designed specifically for the class of *uniform* CTMDPs. In a uniform CTMDP, transitions exhibit all the same exit rate. The algorithm then exploits the uniformity to compute the optimal time-bounded reachability of CTMDPs under the class of time-abstract strategies. The procedure is similar to the original uniformisation [Jen53] for CTMCs. The algorithm was later extended so as to compute resource-bounded reachability in [Bai+08]. It is simple, efficient and easy to implement.

Unif+ [But+15] is a simple and fast algorithm based on *uniformisation* [Jen53] and *untimed* [Bai+05; Brá+13; RS13] analysis. In contrast to the other algorithms the time horizon is not discretised; instead the maximal time-bounded reachability of CTMDPs is iteratively estimated by computing an under and an over approximation. The algorithm can perform in both early and late setting. The computation considers the class of time-abstract strategies. Nevertheless their power tends to the richer class of timed strategies as the *uniformisation rate* tends to infinity. Therefore, in order to improve the approximation bound, in each iteration the uniformisation rate is doubled. The iterations continue until the desired accuracy level is achieved.

Comparison. An broad empirical comparison of the above mentioned algorithms (except for uniformisation and Fu's algorithm) is presented in [But+15] where the efficiency of computing time-bounded reachability for CTMDPs is investigated across several industrial and academic case studies. Although there is no clear winner among the algorithms, except for a few cases, Unif+ and Buchholz-Schulz algorithms stand out. The former provides a simple and fast solution, especially when a low to middle accuracy is required. The latter, however offers a more stable performance for high precision analysis. All in all, Unif+ seems to be the best choice for the first attack to solve the problem.

6.5.2 Resource-to-time transformation

There are a few resource-to-time transformations for Markovian models, which are explained here. A transformation of resource- to time-bounded properties for CTMCs was proposed by Beaudry [Bea78]. It uses rate scaling to compute certain performance-related reliability measures, such as reachability and mean resource before failure. The technique was generalised [Bai+00] and later extended to uniform CTMDPs by [Bai+08] for computing the optimal resource-bounded reachability under the class of time-abstract strategies. We extended this approach in several directions:

(*i*) Our transformation works for two player Markov games [Hat+15], which generalises the previous works. In this thesis, however we focused the transformation on MRAs, which are special cases of Markov games with only one player.

(*ii*) While the original transformation is only given for reachability properties, our extension also supports cumulative and final rewards.

(*iii*) We showed that the transformation is measure preserving under the class of measurable (early) strategies whereas before it was only proved under the class of time-abstract strategies.

(*iv*) In contrast to the previous works, our transformation can deal effectively with resource preserving transitions (transitions with zero resource consumption). This is in fact due to the existence of immediate transitions offered by probabilistic states in MRAs.

6.5.3 Future works

There exist several avenues for improvement and generalisation of our current approach. We start with the algorithm for computing the optimal ETR (Alg. 6.1) and discuss in which direction it can be improved.

As mentioned before, the number of iterations taken by the algorithm is often large. Resorting to higher-order approximations will decrease the number of iterations [HH15; Fea+11; Fea+16]. We have already done this for IMCs [HH15]. To apply this for MRAs, the main challenge is to estimate the swapping points. Technically, employing higher-order approximations enables the optimal strategy to change its decision inside a chunk. This necessitates the estimation of swapping points. To do this in MRAs, we need to consider all stationary strategies for reaching the set of Markovian states from a probabilistic state. This strategy set can in general be exponentially large.

An effective alternative to decreasing the number of iterations is to make the algorithm adaptively taking the discretisation constant according to the current precision achieved. This has been shown to drastically reduce the number of iterations for classical collocation methods, e. g. [Pre+07, Sec. 17.2] and also for CTMDPs [BS11; BDS17]. We believe that our algorithm can be easily improved to make use of *adaptive* discretisation constants.

There is an obvious way to generalise our approach to support a richer class of models. Even though both, the transformation and the algorithm are extended to compute the same measures in two player Markov games [Hat+15], only early strategies are considered there. The extension to the late setting would generalise our approach then effectively cover more formal models including CTMDPs in the late setting.

Chapter 7

Implementation and Case Studies

This chapter revolves around evaluating the effectiveness of the algorithm proposed for the optimal ERR in Ch. 6. It reports on empirical analyses of our implementation over a variety of academic and industrial case studies modelled by MRA. We analyse the models against properties that can indeed be formulated as optimal ERR and are also highly demanded in practice, such as time-bounded reward, resource- and time-bounded reachability.

We explain first how to model large MRAs in an effective way. We then briefly describe IMCA, the tool in which we have implemented our algorithm. Combining both constitutes a processing pipeline called MAMA [Guc+13; Guc+14a; Guc+14b], which not only supports model description from the beginning to the final numerical analysis, but also efficient reduction and minimisation of the model in between.

We demonstrate that the algorithms and tools we developed in the previous chapters are effective and useful in practice. To this end, we have collected, to the best of our knowledge, the richest set of academic and industrial case studies available for MAs and MRAs. We look into the case studies and clarify their components and parts. We evaluate the case studies against different analysis goals. This is followed by experimental results and analysis discussions. In particular, we inspect the sensitivity of our algorithm with respect to certain parameters of the model and the analysis, such as model size and accuracy level.

7.1 MAMA tool chain

In this section we sketch MAMA, a tool chain that will be used for MRA modelling and analysis. MAMA offers an efficient and effective approach that supports all the way from model description to model reduction and evaluation. In this regard, there is a clear separation between “model generation” procedure and “model evaluation” procedure, each being carried out via different tools.

```
1   Producer = <lambda> . tau . psum(0.05 -> Producer[] ++ 0.95 -> send . Producer[])
2   Consumer = receive . <mu> . Consumer[]
3
4   init Producer[] || Consumer[]
5   comm (send, receive, delivered)
6   encap send, receive
```

Figure 7.1: MAPA specification of the producer-consumer depicted in Fig. 1.3

7.1.1 Model generation

MaMa provides a process-algebraic modelling language designed for specifying MRAs. Moreover, it supports several reduction techniques aiming to enhance both model generation and model evaluation.

MAPA. MaMa employs the Markov Automata Process Algebra (MAPA) [Tim13] for specifying MRAs. It supports nondeterminism, probabilistic branching and stochastic timing, all of which are required for specifying MRA models. Moreover, MAPA supports *parallel composition* of MRAs, allowing modular specification of large systems via describing subsystems by MRA and declaring the interaction between them. In addition, it provides several syntactic reductions (see also [Tim+12; TPS13]) applied on the language level that shrink the size of models. MAPA language together with the reduction techniques offer an efficient way to model MRAs and to reduce their size.

An example. To give the flavour of modelling with MAPA, we present the MAPA specification of the producer-consumer example described in Fig. 1.3. It is shown in Fig. 7.1. In the specification random delays are enclosed in "<>", actions are written in red colours and probabilistic branching is represented by the psum operator. The MAPA specification contains two processes Producer (line 1) and Consumer (line 2). The Producer is subject to a random delay with rate lambda for producing an item. The item is either failed to be produced with probability 0.05, or is successfully produced and sent to the Consumer with probability 0.95 via action send. The Consumer receives the item via action receive and consume it in a period that is exponentially distributed with rate mu.

MAPA provides certain constructions for specifying how processes interact with each other. The init statement on line 4 indicates that the Producer and the Consumer run in parallel. On line 5, comm describes the action synchronisation between the two processes. It states that *send* is synchronised with receive; the result is labelled with delivered. Afterwards actions send and receive are made invisible. This is done in line 6 using the encap statement.

SCOOP. Support for MAPA specifications together with several reduction technique is implemented in SCOOP [Tim+12]. SCOOP takes a MAPA specification, applies the reduction techniques at the language level and finally generates the resulting model. The final model basically describes the states and the transitions in between. It can be dumped into specific formats that can be read by different analysis tools, including IMCA, which is our tool for MRA analysis. A web service offering, among others, SCOOP functionality is available at `http://wwwhome.cs.utwente.nl/~timmer/scoop/webbased.html`. Detailed discussions about MAPA and SCOOP are the subject of Ph. D. thesis of Mark Timmer [Tim13].

7.1.2 Model evaluation

This section elaborates on our implementation that computes an approximation of the optimal ERR. We first discuss interactive Markov chain analyser (IMCA), the part of MAMA that is responsible for MRA analysis.

```
#INITIALS
s0
#TRANSITIONS
s0 ! 0.0
* s1 3.0
s1 tau
* s1 0.05
* s2 0.95
s2 send
* s0 1.0
```

Figure 7.2: An IMCA format

IMCA. The IMCA tool is the part of MAMA tool chain targeted at efficient computation of various types of objectives for MRA models. We have implemented Alg. 6.1 in IMCA [Guc12; Guc+12; Guc+13; Guc+14b; Guc+14a]. It accepts the textual representation of MRAs in the form of state-transition list, resembling the input language of MRMC [Kat+09] and ETMCC [Her+03]. The IMCA format of the producer corresponding to line 1 of Fig. 7.1 and also to Fig. 1.3a is illustrated in Fig. 7.2. Notice that the IMCA format precisely follows the state-transition graph of the MRA. The initial state is described by `#INITIALS`. The declaration of transitions are followed after `#TRANSITIONS` statement. A transition is specified by its source state followed by an action name and a (rate) distribution. Special action `!` is used to signify a Markovian transition.

The block diagram of IMCA is depicted in Fig. 7.3. The parser reads a model specified in IMCA format and translates it into a specific sparse matrix equipped with several indices. The indices provide a fast access to the model parameters such as rates, distributions and rewards of the transitions. This data structure is similar to the one used by MRMC. For more details about the data structure see [Guc12].

The analysis engine of IMCA includes a diverse range of analyses. The "reachability engine" of IMCA computes the optimal unbounded and timed reachability probabilities of MRAs. The optimal long-run average (reward) is carried out by the "long-run engine". The "reward engine" offers reward computation under both time-bounded and goal-bounded restrictions. The latter stands for the optimal expected reward gained until reaching certain goal states. The former refers to the optimal ETR. Our contribution to the tool was made by providing the engine for computing timed reachability and the optimal ETR.

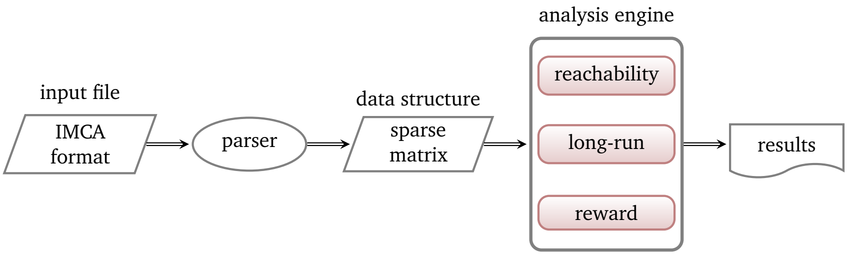


Figure 7.3: The IMCA tool [Guc12; Guc+12; Guc+13; Guc+14b; Guc+14a]

Implementation details. To analyse an MRA against the optimal ERR it might be required to apply the resource-to-time transformation before feeding the model into Alg. 6.1. We have implemented the transformation using a Python script. The transformation is done at the language level, i. e. both the original and the transformed model are described in IMCA format. The complexity of the transformation is linear in the size of the model and can be therefore carried out in an efficient way.

We now remark on our implementation of Alg. 6.1. The crucial part is to perform the P-updates, which is done by function `Pupdate` in the algorithm. There, the least fixed point of the Bellman optimality operator given in Eq. (6.38) is computed. There are different ways to compute the fixed point. A simple approach, which is also implemented by us, is to use *value iteration*. This approach iteratively updates the vector of rewards using the Bellman optimality operator. At the beginning the vector is initialised to zero, i. e. $\boldsymbol{g} = \mathbf{0}$. The iterations continue until a stopping criterion is met. A common stopping criterion is to terminate the iterations if the distance between two successive valuations is less than some predefined error tolerance.

For selecting a good discretisation constant that respects the given accuracy level (line 9 of Alg. 6.1) we use both linear approximation, given by Eq. (6.39) and Newton steps. Unless stated differently, in the conducted experiments the Newton method is applied as it leads to a fewer number of iterations. For sensitivity analysis of the algorithm however we used the linear approximation. The optimal ERR, which is the main analysis discussed in this thesis, embodies various types of objectives such as time- and resource bounded reachability and time-bounded cumulative reward. In Ch. 6, Alg. 6.1 is proposed to compute ϵ-approximation of the optimal ETR using the discretisation technique described in Sec. 6.2. Computing the optimal ERR reduces to computing the optimal ETR via the transformation given in Def. 6.22.

7.1.3 The tool chain

In order to enable the analysis of MRAs, SCOOP has been connected to IMCA to form a tool chain, called MAMA [Guc+13; Guc+14a]. In its middle, however we will need to apply the transformation described in Def. 6.22, in case a resource-

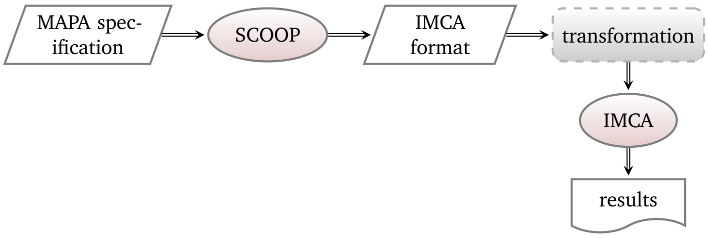


Figure 7.4: The MAMA tool chain

bounded analysis is asked. The process of MRA analysis using the tool chain is summarised in Fig. 7.4. Once an MRA is specified in MAPA, the specification is fed into SCOOP. SCOOP then performs various reductions and minimisations and finally generates the model directly in IMCA format. The IMCA format can now be analysed against time-bounded properties, e. g. the optimal ETR. Nevertheless, we need to apply the transformation before the model is handed over to IMCA, provided the analysis involves some resource-bounded property. In this case the script translates the model into its transformed version according to Def. 6.22.

As mentioned before, this tool chain is not restricted to ETR and ERR analysis. Once the model is entered, IMCA in addition can carry out long-run average rewards, unbounded reachability and expected goal-bounded rewards. A web interface for MAMA is available at `http://wwwhome.cs.utwente.nl/~timmer/scoop/webbased.html`.

7.2 Case Studies

For our experiments we use a diverse collection of industrial and academic case studies, all modelled as MRA. Their size is ranging from small models to very large models containing hundreds of thousands states. Each case study comprises several components communicating with each other. We model the components and their interactions using MAPA. In what comes next, we describe different components of each case study and explain how they are assembled together to construct the whole system.

Polling System [Guc+13; TPS13] consists of S stations and one server, as illustrated in Fig. 7.5. Incoming requests of J types are buffered in the stations, each equipped with a queue of size K. The server polls the stations for requests and processes them with a rate depending on their type. It may happen with probability 0.1 that a request erroneously stays in a queue even after being processed by the server. This is represented by red arrows in the figure. Each request brings an instantaneous reward once it is completely processed by the server. Moreover, when processing, the server consumes energy. The model is subject to two kinds of analysis: First

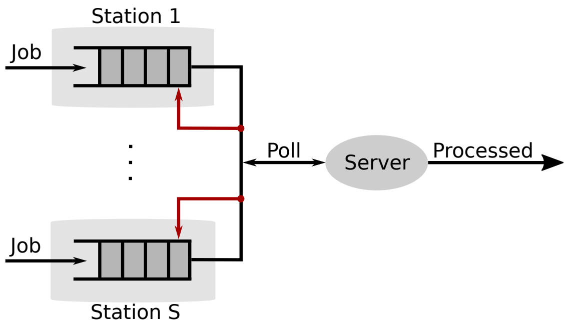


Figure 7.5: Diagrammatic representation of a polling system

we compute the minimum and the maximum probability of encountering the error under some energy budget. The second analysis is on the computation of the minimum and the maximum expected energy-bounded reward of the model. The instances of the polling system are denoted by "PS-S-J-K".

Queuing System [HH12] stores requests of J different types into two queues, each of size K. Each queue is attached to a server, as depicted in Fig. 7.6. Servers A and B fetch requests from their corresponding queues and process them afterwards. Requests processed by server A may follow different paths. They are either considered as "completely done" or are subject to "further processing" by server B. In the latter case, which happens with probability 0.1, they are sent back to the queue of server B as illustrated by the blue arrow in the figure. Both servers consume energy when they are processing. We compute the minimum and the maximum number of processed requests under different energy budgets. The model instances are denoted as "QS-J-K".

Dynamic Power Management System [QWP01] models the power management system used in a model of Fujitsu disk drive. An abstract model of the system is described in Fig. 7.7. The model consists of four compo-

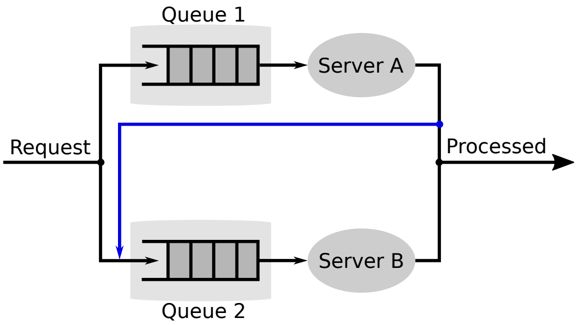


Figure 7.6: Diagrammatic representation of a queuing system

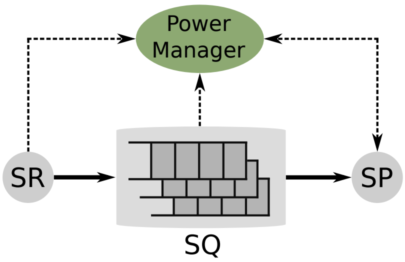


Figure 7.7: The abstract model of the dynamic power management system

nents: service requester (SR), service queue (SQ), service provider (SP), and power manager (PM). SR generates tasks of J types differing in energy demands. Tasks are buffered by SQ, providing a separate queue of size K for each task type. They are then fetched from the queues and delivered to SP to be processed. SP can work in different energy/performance levels ranging from sleep and stand-by to full processing mode. It is the task of PM to control the mode of SP in order to achieve a specific objective, like minimising the energy consumption or processing time.

We explain in more details the operation modes of the SP. It can either be "busy" with processing a task, "idle" while the queues are empty, in a "standby" mode, or in a "sleep" mode. In the latter two modes it is inactive and unable to handle tasks. The change between "busy" and "idle" occurs automatically, depending on whether there are tasks in the queue or not. If it has been "idle" for some time, it can be switched into "standby" or "sleep" by PM. The PM is also responsible for switching from these two modes back to "idle". The SP consumes the least power in "standby" and "sleep" (0.35 W and 0.13 W, respectively), whereas it consumes more power while "idle" (0.95 W) and the most if it is "busy" (2.15 W) [QWP01; Sim+00]. We model the DPMS as an MRA with the resource representing the power consumption of the SP. The reward granted as a task is completely accomplished. For our experiments we varied the number of different task types (J) and the size of the queues (K) and describe the instances by "DPMS-J-K". We explore the optimal ERR under different energy budgets.

Stochastic Job Scheduling [BDF81] originally stems from economy. In this setting, a number of jobs with different service rates are distributed between processors. Each processor consumes resources, e. g. energy which has to be paid for. The goal is to have all jobs processed within a certain energy budget. In our experiments we explore the reachability of this goal with homogeneous ("all processors have the same power consumption") and heterogeneous power consumption ("the power consumption is different across processors"), while varying the number of jobs (M) and the number of processors (N). Since the system degenerates to a CTMC if the service rates are homogeneous, we do not consider this case. The model instances

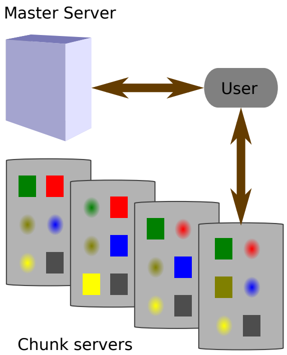


Figure 7.8: The google file system

are denoted as “SJS-N-M”.

Google File System [GGL03; Guc12] is a model of a replicated file system that is used by Google, as depicted in Fig. 7.8. It splits files into chunks of equal size. A copy of each chunk is maintained independently by several of N chunk servers. In the figure, copies of a chunk in different servers are represented by the same colour. If a user requests a read/write access to a chunk, the file system first asks a master server which stores the addresses of all individual chunks. Afterwards, the data is transferred between the file system and the appropriate chunk server. All servers are prone to hardware and software failures, but they can be repaired in case a failure happens. Moreover, a copy of a chunk may be lost (illustrated by blurred colour), however the recovery is possible if the master server is up. The number of chunks a server may store is fixed to 5000 and the total number of chunks to 100000. The model instances are then produced by varying the number of chunk servers N, denoted as “GFS-N”. We analyse the maximal reachability probabilities of several instances of GFS. A state is defined as goal if the file system service is operational there, i. e. the master server is up and for each chunk at least one copy is available. We therefore estimate the probability of the system to become operational within some time interval if starting from a state at which the system is out of service.

Fault Tolerant Workstation Cluster [HHK00], originally described as a GSPN, models two networks of N workstations each, interconnected by a switch. The two switches communicate via a backbone, as depicted in Fig. 7.9. Workstations, switches, and the backbone fail after a random time that is exponentially distributed, and can be repaired only one at a time. If multiple components have failed at the same time, a choice needs to be made what to repair next. The model instances are denoted by “FTWC-N”. We consider the overall cluster in premium quality if at least N workstations are operational and connected to each other. In our experiments, we

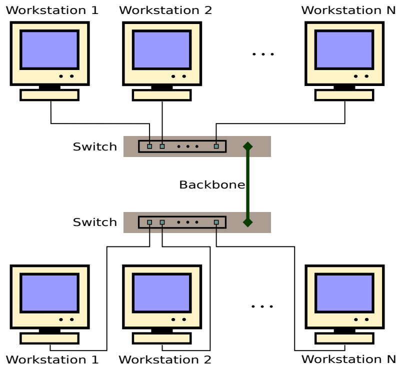


Figure 7.9: The fault-tolerant workstation cluster

analyse the probability of the cluster being unable to offer the premium service within specific periods of time.

Erlang Stages is a synthetic model with known characteristics [ZN10]. It has two different paths to reach the goal state: a fast but risky path or a slow but sure path. The slow path is an Erlang chain of length K and rate R. The model instances are produced by varying the length K and the rate R of the slow chain (Erlang chain). ES-K-R

7.3 Results and discussions

In this section we report on the experimental results of analysing the case studies discussed in Sec. 7.2 against various kinds of objectives. As we mentioned before, we exploit the MAMA tool chain, described in Sec. 7.1, for this analysis process. To do this, we made the MAPA specification of each case study. We then fed the MAPA specifications into SCOOP to generate the IMCA format. The IMCA format is subject to the resource-to-time transformation if the property to be analysed is in resource-bounded setting. The IMCA format can then directly be analysed using IMCA. Here we provide the results of the analyses and discuss them in more details.

Experimental settings. We ran our experiments on two different machines. All experiments on the first four case studies were run on a single core of an Intel Xeon processor (quad-core, 3.3 GHz per core) with 64 GB of memory. All others were run on a single core of Intel Core i7-4790 (quad core, 3.6 GHz per core) with 16GB of RAM. In the sequel, we report the parameters of the model we analyse, the objective to be computed and the result of the computation. We sometimes report the duration of the analysis, but not the memory consumption, since it is negligible for all experiments. This echoes that the space complexity of Alg. 6.1 is linear in the model size.

Table 7.1: Expected reward in the dynamic power management system

		budget = 10		budget = 20		budget = 50	
name	#states	min	max	min	max	min	max
DPMS-2-5	508	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-2-10	1,588	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-2-20	5,548	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-3-5	5,190	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-3-10	29,530	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-3-20	195,810	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-4-5	47,528	0.784	0.877	1.617	1.889	4.143	4.936
DPMS-4-10	492,478	0.784	0.877	1.617	1.889	4.143	4.936

Computation times. Here we briefly make some observations on the time measurements of all experiments. The shortest computations took only fractions of a second, e. g. the computation of the minimum reachability for SJS-2-4 with a resource budget of 5 took 0.06 seconds. The larger ones, however needed several hours, e. g. for DPMS-4-10 the computation of the minimum reachability with a resource budget of 50 took almost 11 hours, which was the highest computation time across all our experiments. As expected, larger models need more time for analysis than smaller ones. The computation time is also influenced by the size of the resource (time) budget. For example, for resource budget 10 instead of 50, the computation of the minimum reachability for DPMS-4-10 took less than 6 min. Other parameters that affect the running time of Alg. 6.1 are the accuracy level, the largest exit rate and the maximum reward (of different kinds) appearing in the model. We discuss in detail later how these parameters affect the computation time.

Empirical results and discussions. At this point we present our empirical observations. Tab. 7.1 to 7.4 show the results of our experiments for the first four case studies, one table for each case study. The first two columns of each table contain the name of the respective model instance and its number of states. We reported the result of our computations for different kinds of objectives.

In case of DPMS (Tab. 7.1) and QS (Tab. 7.2) we explore the minimum and maximum expected reward under different resource budgets. For DPMS we used resource budgets of 10, 20, and 50, whereas for QS we used resource budgets of 1, 5, and 10 (see the respective blocks in Tab. 7.1 and Tab. 7.2). It holds for both DPMS and QS that the expected reward grows with the budget, as does the difference between minimum and maximum reward, as to be expected. Another interesting fact is that the size of the queues in the models – while having a considerable influence on the size of the system – has practically no impact on the expected reward. The behaviour is completely determined by the number of different task types. This observation can be explained as follows: For the processing unit of DPMS (or of QS) it is not important how many jobs exactly can be stored in the queue(s), as long as there *are* jobs in the queue(s).

For PS (Tab. 7.3) we studied both the minimum and maximum reachability and the minimum and maximum expected reward (see the respective blocks in the table) under a resource budget of 5. If we increase the queue size, the minimum and maximum probability for encountering the error decreases, while the expected minimum and maximum reward increases. At the same time we can observe that the reachability increases with the number of stations, e. g. for PS-2-2-2, containing two stations, the maximum probability is 0.773, whereas for PS-5-2-2, containing five stations, it is 0.992. This makes sense, since the error is caused by the stations and the probability to encounter the error therefore increases with having more stations.

For SJS (Tab. 7.4) we also used a resource budget of 5. Here we studied the minimum and maximum reachability while assuming homogeneous or heterogeneous energy consumption for the different processors of the system (see the respective blocks in Tab. 7.4). For homogeneous energy consumption we can observe a similar effect as for DPMS and PS: The number of processors influences the number of states in the system, but has a negligible impact on the reachability. The latter is completely determined by the number of jobs. What's more, the minimum and the maximum reachability are the same in this case, which echoes the fact that the nondeterminism there is spurious. These effects vanish if we assume heterogeneous energy consumption. In this case, the distance between the minimum and maximum reachability increases, especially the maximum reachability becomes higher. These observations make sense: In case of a homogeneous system it does not matter, which processor handles which job. However, in a heterogeneous system there is a choice between more and less expensive processors which can handle the jobs, which in turn leads to a higher

Table 7.2: Expected reward of the queuing system

		budget = 1		budget = 5		budget = 10	
name	#states	min	max	min	max	min	max
QS-2-2	2,314	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-3	10,778	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-4	46,234	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-5	191,258	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-6	777,754	0.249	0.857	1.294	4.078	2.634	7.975
QS-3-2	12,205	0.125	0.857	0.649	4.078	1.332	7.972
QS-3-3	117,532	0.125	0.857	0.649	4.078	1.332	7.972
QS-3-4	1,080,865	0.125	0.857	0.649	4.078	1.332	7.972
QS-4-2	42,616	0.125	1.287	0.649	6.127	1.333	12.075
QS-4-3	708,088	0.125	1.287	0.649	6.127	1.333	12.075
QS-6-2	266,974	0.084	1.713	0.433	8.187	0.892	16.201

Table 7.3: Results for the polling system

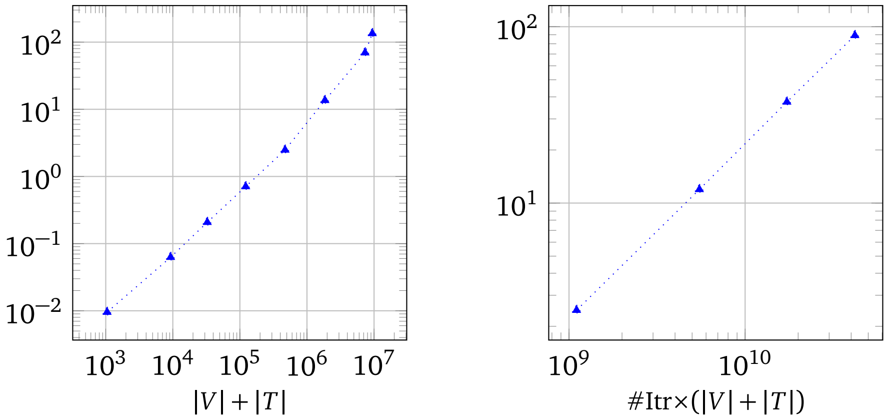
		rechability		reward	
name	#states	min	max	min	max
PS-2-2-2	455	0.743	0.773	3.128	3.219
PS-2-2-3	2,055	0.483	0.551	3.980	4.117
PS-2-2-4	8,421	0.998	0.999	1.045	1.080
PS-2-3-2	2,392	0.995	0.996	1.209	1.253
PS-2-3-3	22,480	0.973	0.983	1.730	1.848
PS-2-3-4	137,445	0.990	0.994	1.489	1.583
PS-3-2-2	3,577	0.888	0.917	2.549	2.685
PS-3-2-3	34,425	0.665	0.760	3.493	3.732
PS-3-3-2	35,659	1.000	1.000	0.918	0.965
PS-3-4-2	300,793	0.402	0.543	4.180	4.412
PS-4-2-2	27,783	0.955	0.973	2.166	2.307
PS-4-3-2	570,375	0.793	0.879	3.116	3.403
PS-5-2-2	213,689	0.983	0.992	1.908	2.039

Table 7.4: Reachability in the stochastic job scheduling benchmark

		homogeneous costs		heterogeneous costs	
name	#states	min	max	min	max
SJS-2-2	34	0.713	0.713	0.699	0.799
SJS-2-4	464	0.241	0.241	0.186	0.243
SJS-2-6	4,144	0.041	0.041	0.021	0.029
SJS-2-8	29,344	0.004	0.004	0.001	0.002
SJS-4-2	104	0.713	0.713	0.542	0.995
SJS-4-4	3,168	0.241	0.241	0.120	0.610
SJS-4-6	71,644	0.041	0.041	0.013	0.130
SJS-4-8	1,032,272	0.004	0.004	0.001	0.012
SJS-6-2	214	0.713	0.713	0.424	1.000
SJS-6-4	13,924	0.241	0.241	0.059	0.945
SJS-6-6	685,774	0.041	0.041	0.005	0.374
SJS-8-2	364	0.713	0.713	0.337	1.000
SJS-8-4	41,552	0.241	0.241	0.033	0.999
SJS-10-2	554	0.713	0.713	0.274	1.000
SJS-10-4	98,436	0.241	0.241	0.019	1.000

(lower) maximum (minimum) reachability.

We now investigate specific parameter changes and their impact on the running time of the algorithm. We begin with the model size, more specifically, the number of states and transitions in the model. By looking into Alg. 6.1, it be-



(a) Fault tolerant workstation cluster

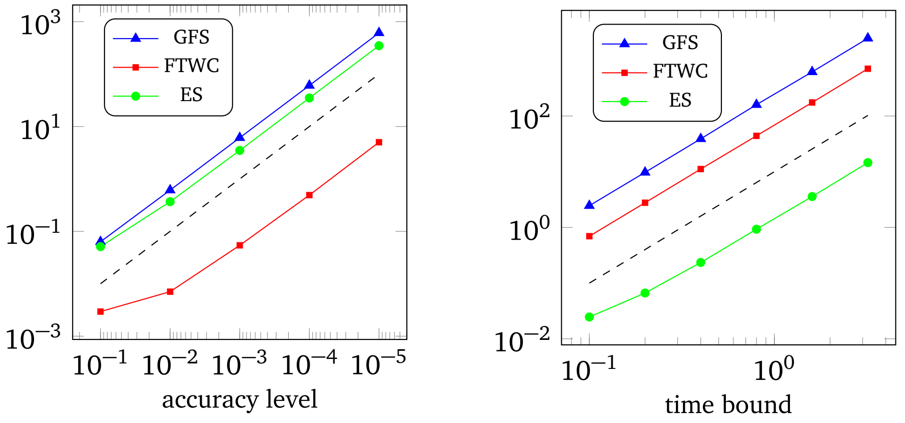
(b) Google file system

Figure 7.10: Impact of model size on computation time

comes clear that enlarging the model size increases the per iteration complexity of the algorithm. It affects both M- and P-updates in the same way. On the one hand, if the number of Markovian states and/or transitions increases, M-updates will need to perform more operations. The same situation prevails among the probabilistic part of the model when a P-update is invoked. In this case, the fixed point is computed over a larger structure. Since we employ value iteration in our implementation, we can say that the per iteration complexity of Alg. 6.1 is proportional to the model size.

We inspect this claim in practice on the Google file system and the fault tolerant workstation cluster analysed against time-bounded reachability. We fix the accuracy level and the time bound and generate several models by changing their other parameters. This gives us a bunch of similarly structured models that differ in size. Unfortunately it is not always possible to keep the maximum exit rate fixed across the models. This impacts the number of iterations to be taken, which directly affects the running time. We observe this phenomenon in Fig. 7.10a. In this figure the number of states plus the number of transitions is considered as the representation of the model size for the fault tolerant workstation cluster. On the y-axis the running time of the algorithm is reported. The plot resembles a line with slope of one in logarithmic scale, which reflects the fact that the running time is proportional to the model size.

The last sample observed from Fig. 7.10a deviates from the previous sample paths. The reason, as we mentioned briefly, is that the corresponding model has a larger maximum exit rate than the others. This makes the algorithm take more iterations to respect the given accuracy level. This situation happens more frequently for the Google file system. We therefore take the number of iterations into account and multiply it to the model size. We then report the running time of the algorithm for each model instance as shown in Fig. 7.10b. Owing



(a) Impact of accuracy on running time (b) Impact of time bound on running time

Figure 7.11: Impact of accuracy level and resource budget on computation time

to considering the number of iterations, the high deviation disappears in this case. Hence, we shall investigate the number of iterations and its impact on the running time.

The computation time of Alg. 6.1 also depends on the size of the resource budget[1] and the required accuracy level. In contrast to model size which affects the per iteration complexity of the algorithm, these two parameters influence the number of iterations that need to be taken in order to satisfy the accuracy level. To see why, notice that the per iteration complexity is solely influenced by the size and the structure of the model, i. e. the number of states and transitions and how they are connected. Hence, the aforementioned parameters have no impact on the per iteration complexity. Moreover, considering Eq. (6.30) makes clear that raising the accuracy level or extending the resource budget both increase the number of iterations of the algorithm. The increasing ratio is however different for the two parameters. Extending the resource budget by factor α with other model parameters unchanged increases the number of iterations by factor α^2. The corresponding factor for the accuracy level under the same condition is α. Note that the observations are only valid if we use the linear approximation given in Eq. (6.35) to estimate the number of iterations of the algorithm (see also Alg. 6.1 line 9).

We conducted some time-bounded reachability analyses in order to find out the impact of changes in time bound and accuracy level on the running time of our algorithm. Even though an estimation of the discretisation step via the Newton method always gives better results than using the linear approximation, for our purpose we take the latter in this set of experiments. The running time of the algorithm with respect to ϵ and b are illustrated in logarithmic scale in

[1] In this context we can indistinguishably interchange between resource budget and time bound as the transformation defined in Def. 6.22 does not modify its size.

Fig. 7.11a and 7.11b, respectively. The dashed lines plotted in each figure are meant to help representing the dependency. They are drawn with the slope of one and two in the left and the right figure, respectively. As it can be seen, the observation from empirical results justifies the conclusion made in the above paragraph. They suggest the linear dependency between running time and ϵ and quadratic dependency between running time and and b.

There are in general other parameters affecting the number of iterations. Their impact on the number of iterations is summarised in Sec. 6.3.2 by Eq. (6.40). From there it can be observed that for time-(resource-)bounded reachability ($L^c = 0$) the number of iterations is quadratic in the maximum exit rate. Other dependencies can be inferred from Eq. (6.40).

7.4 Discussion

In this chapter we elaborated on the details of the analysis process we used for computing the optimal ERR for MRAs. Furthermore, we reported on experimental results of the analysis on several case studies. The main points covered in this chapter are as follows.

(*i*) We discussed briefly how to use Markov automata process algebra (MAPA) for modelling MRAs;

(*ii*) We explained the implementation of the analysis technique for computing the optimal ERR, which is mainly as a part of IMCA tool;

(*iii*) We described the MAMA tool chain, which provides an analysis pipeline for computing various kinds of objectives defined on MRAs;

(*iv*) We collected several industrial and academic case studies that are all modelled by MRA;

(*v*) We conducted various experiments on the set of case studies to enquire how effective and efficient our analysis methodology is. Moreover, we studied the runtime sensitivity of our algorithm with respect to different parameters of the model and the objective.

Contributions. This chapter contributed to the practical applications of MRAs and provided an effective and solid methodology for their analysis. This is done via the MAMA tool chain [Guc+13; Tim13; Guc+14a; Guc+14b], constituting a pipeline for modelling, reduction and analysis of MRAs. Research groups from Universität des Saarlandes, RWTH Aachen and University of Twente were involved in the development of the tool chain. Our contribution in this collaboration was the development of algorithms and tools for MRAs analysis against the optimal ERR.

Markov (reward) automaton is a relatively new model with various application areas. However, the number of case studies and benchmarks that demonstrates their applicability has been thus far very limited. In this chapter, we collected several case studies from industry, economics and academia to better demonstrate the potential application domains of MRAs. Except for the Google file system for which the IMCA format was available [Guc12], we specified all the other case studies in MAPA.

Related works. The root of our analysis techniques mainly comes from the CTMDP world. A number of tools have been developed for the analysis of CTMDPs and related models. ETMCC [Her+03] was one of the first tools for analysis of continuous-time Markovian models. It supports model checking of CTMCs against continuous stochastic logic (CSL). The successor of this tool, MRMC [Kat+09] provides a wide range of analysis on various kinds of models. In particular, it supports CTMDPs and their model checking. There, the Buchholz-Schulz algorithm [BS11; Buc+11] is implemented for computing the optimal ETR of CTMDPs. IMCA is highly inspired by MRMC. It reuses MRMC data structure with slight adaptions to MRAs. However, as we discussed in Sec. 6.5 the algorithm for computing the optimal ETR follows a different approach. There exists an algorithm [Fu14a; Fu14b] for resource-bounded reachability of CTMDPs with multi-dimensional resource consumption. Nevertheless this algorithm has not been implemented in any tool and thus far has been known as a proof of concept for solving the problem (on multi-dimension). To the best of our knowledge there is neither a tool that employs the polynomial approximation [Fea+11; Fea+16] nor a prototype implementation of the algorithm. In the IMC world, the Zhang-Neuäusser algorithm was proposed for computing time-bounded reachability. The algorithm is implemented in the reachability engine of IMCA. As discussed in Sec. 6.5 our analysis technique utilises the same approach but for a more general class of problems on a more general model.

Future works. There is potential room for improvement of our implementation of computing the optimal ETR. Notably, the discretisation can be extended to take adaptive steps instead of fixed steps. This has been shown to bring considerable advancement in efficiency for conventional ODE solvers [Pre+07, Sec. 17.2], and also for the Buchholz-Schulz algorithm [BS11; But+15]. Another improvement may result from employing a more efficient approach for computing P-updates. Currently P-updates are computed via value iteration. In addition, *policy iteration* and *linear programming* may be used to compute the fixed point. An empirical comparison of these approaches given in [For+11] shows that in all of their benchmarks a variant of value iteration stands out. However, there are situations where a combination of the approaches is more effective. This is the case, for instance when the exact value of the fixed point is needed. For this problem, it has been shown [Gir12] that the combination of value iteration and linear programming is more efficient. There are plenty of other possibly more efficient approaches, e. g. pre-solving of P-updates, topological value iteration [DG07] and so on.

Chapter 8

Conclusions

This thesis has contributed to the analysis of *Markov (reward) automata* (MRA). MRAs constitute an expressive and powerful formalism encompassing a wide range of features that are required for design and analysis of complex systems. In order to develop an effective and sound analysis framework for MRAs, we studied their theoretical foundations and concepts. We then considered their analysis on finite horizon and, more specifically studied the optimal expected resource-bounded reward (ERR). In addition to be challenging, this embodies a collection of important analyses, which are naturally needed for evaluation of industrial and practical systems. We then looked into mathematical characteristics of the optimal ERR. Afterwards, we developed sound and efficient algorithmic solutions for computing the optimal ERR. We implemented our analysis framework and demonstrated its effectiveness on different industrial and academic case studies.

8.1 Summary

We summarise the main topics explored in this thesis chapter by chapter.

Ch. 1 started with a brief introduction to MRAs and indicated that MRAs can serve as an effective modelling formalism for real world systems. It then discussed why the optimal ERR is essential for analysis of practical systems.

Ch. 2 provided the mathematical background essential for understanding the concepts, methods and proofs used in this thesis. This includes measure theoretic concepts, convergence, continuity and basics of probability theory.

Ch. 3 developed the theoretical foundations of Markov automata (MA), the underlying construction of MRAs. It defined Markov automaton and discussed compositionality. It then explained the concept of *closedness* and provided the semantics of *closed* Markov automata by means of *histories* and *paths*. Afterwards, it studied measurability in the MA context, and defined a σ-algebra over the set of paths induced by MAs together with its *generic measurable strategies*. The most important topic of this chapter was *expectation splitting*, which provides a generic technique for computing the expectation of a random variable defined on the set of paths induced by an MA.

Ch. 4 introduced Markov reward automata (MRA) as the extension of Markov automata with reward and resource structures. It subsequently defined the optimal *expected resource-bounded reward* (ERR) on MRAs, which is the main objective studied in this thesis. As the main result, this chapter established an upper bound for the optimal ERR of an MRA in terms of certain parameters of the model. Accordingly, MRAs were classified into *null reward convergent* and *null reward divergent*, depending on the value of the optimal ERR being finite or infinite, respectively.

Ch. 5 took the main step toward computing the optimal ERR by characterising it as the least fixed point of a Volterra integral equation system. Using this result, it established Lipschitz continuity of the optimal ERR for null reward convergent models. This also leads to its differentiability *almost everywhere* but not at *swapping points*.

Ch. 6 addressed the numerical computation of the optimal ERR. It started with a slightly simpler problem, namely the computation of the optimal *expected time-bounded reward* (ETR), which is the optimal ERR when time takes the role of the resource. For this, a sound discretisation approach was proposed for computing an ϵ-approximation of the optimal ETR. This leads to a numerically stable algorithm with strict a priori error bound which runs in polynomial time. This chapter also proposed an efficient resource-to-time transformation that reduces the computation of the optimal ERR to that of the optimal ETR. This accomplished a sound analysis framework for computing the optimal ERR in MRAs.

Ch. 7 evaluated the effectiveness and the efficiency of the proposed analysis framework. It first explained our tool chain, MaMa, which constitutes a processing pipeline for efficient modelling, reduction and analysis of MRAs. MaMa is the result of collaboration between a number of research groups from Universität des Saarlandes, RWTH Aachen and University of Twente. Moreover, this chapter collected several industrial and academic case studies that are all modelled by MRA. It then reported on an empirical evaluation of our analysis framework by conducting various experiments on the set of case studies. Moreover, the run-time sensitivity of our technique was studied with respect to different parameters of the model and the objective.

In summary, the framework proposed in this thesis provides sound and efficient algorithms for analysing MRAs on finite horizon. This is bundled into the MaMa tool chain together with a modelling language, several reduction techniques and infinite horizon analysis. All in all, MRAs can effectively serve as modelling formalism accompanied by efficient analysis approaches for a wide range of systems.

8.2 Contributions

We categorise the contributions made by this thesis into three domains. We elaborate on each domain separately.

Foundation. This thesis contributed to mathematical foundations of Markov reward automata. In Ch. 3, we extended our previous work [HH12] on the measurability in the MA context, defined the class of *generic measurable strategies* and established a unique probability measure on the set of paths induced by an MRA. Such a thorough study has not been done before for MRAs and MAs. Moreover, we introduced the concept of expectation splitting, an abstract tool that can simplify a diverse range of analyses on MRAs. In Ch. 4, we generalised our previous works [HH12; Guc+13; Guc+14b; Guc+14a; Bra+15; Hat+15] and defined the optimal expected resource-bounded reward (ERR), step by step using the optimal expectation of a sequence of random variables. Employing such a method for defining a measure, later facilitated the development of mathematical features like measurability and continuity. In contrast to common belief, we also showed that studying the *minimal* ERR is technically more difficult than the *maximal* case. While most of the previous works restricted their analysis to the maximal case, we developed our theory for both.

Analysis. A significant contribution was made in the domain of developing both theoretic and algorithmic tools for analysis of Markov reward automata on finite horizon. In Ch. 4, we established an upper bound for the optimal ERR of MRAs. To this end, we utilised the rigorous technique of expectation splitting developed in Ch. 3. Accordingly, we classified MRAs into *null reward convergent* and *null reward divergent*, depending on the value of the optimal ERR being finite or infinite, respectively. In Ch. 6, we developed a sound discretisation approach with strict error bound that leads to a polynomial time algorithm for computing the optimal ETR. The algorithm enables to approximate the optimal ETR up to a given precision level. This is a generalisation of our previous works [HH12; Guc+13; Guc+14a; Guc+14b; Bra+15]. We then proposed an efficient resource-to-time transformation that reduces the problem of computing the optimal ERR in an MRA to the problem of computing the optimal ETR in the transformed MRA. This contribution was published in [Hat+15]. At the end, the algorithm and the transformation together constitute an efficient and sound analysis framework for computing an ϵ-approximation of the optimal ERR.

Application. In this thesis, we also looked into application domains of Markov reward automata. In Ch. 7, we collected several case studies from industry, economics and academia to better show the potential application domains of MRAs. Such a broad list of case studies was never collected for MRAs before. We also implemented our analysis technique as a part of the MAMA tool chain. In addition to our analysis framework, MAMA provides a wide range of tools and techniques for efficient modelling, reduction and analysis (on infinite horizon) of MRAs. We then conducted various experiments on the collected case studies to demonstrate the effectiveness and the efficiency of our analysis framework.

8.3 Future works

A number of topics have been mentioned as future works throughout this thesis. Here we highlight some interesting ones. In Ch. 3, we would like to inspect the existence of a super class of generic measurable strategies that still leads to a unique probability measure over the set of paths induced by an MRA. The result is of theoretical interest. A detailed study of *null reward divergence*, defined in Ch. 4 and its connection to *timelock* is another potential area for further investigation. Establishing the differential characterisation of the optimal ERR, as mentioned in Ch. 5, might provide new insights into the way we analyse MRAs. Moreover, this might be helpful when studying the existence and the shape of the optimal strategy for ERR. The main area of development in Ch. 6 and 7 is to improve the efficiency of Alg. 6.1, which computes an approximation of the optimal ETR. This includes decreasing the number of iterations in the algorithm, developing adaptive discretisation, using more efficient ways for fixed point computation and employing parallel processing. Extending the analysis framework to the late setting will also generalise our approach to a wide range of models, including CTMDPs.

Bibliography

[AD99] Robert B. Ash and Catherine A. Doléans-Dade. *Probability & Measure Theory*. 2nd. Academic Press, 1999.

[AHK03] Suzana Andova, Holger Hermanns, and Joost-Pieter Katoen. “Discrete-Time Rewards Model-Checked”. In: *formats*. Vol. 2791. LNCS. Springer, 2003, pp. 88–104.

[Bai+00] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. “On the Logical Characterisation of Performability Properties”. In: *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*. 2000, pp. 780–792.

[Bai+03] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. “Model-Checking Algorithms for Continuous-Time Markov Chains”. In: *IEEE Trans. Software Eng.* 29.6 (2003), pp. 524–541.

[Bai+05] Christel Baier, Holger Hermanns, Joost-Pieter Katoen, and Boudewijn R. Haverkort. “Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes”. In: *Theor. Comput. Sci.* 345.1 (2005), pp. 2–26.

[Bai+08] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. “Reachability in continuous-time Markov reward decision processes”. In: *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*. 2008, pp. 53–72.

[BCS10] Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga. “A Rigorous, Compositional, and Extensible Framework for Dynamic Fault Tree Analysis”. In: *IEEE Trans. Dependable Sec. Comput.* 7.2 (2010), pp. 128–143.

[BDF81] John L. Bruno, Peter J. Downey, and Greg N. Frederickson. “Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan”. In: *J. ACM* 28.1 (1981), pp. 100–113.

[BDS17] Peter Buchholz, Iryna Dohndorf, and Dimitri Scheftelowitsch. "Optimal decisions for continuous time Markov decision processes over finite planning horizons". In: *Computers & Operations Research* 77 (2017), pp. 267–278.

[Bea78] M. Danielle Beaudry. "Performance-Related Reliability Measures for Computing Systems". In: *IEEE Trans. Computers* 27.6 (1978), pp. 540–547.

[Bin82] K.G. Binmore. *Mathematical Analysis: A Straightforward Approach*. Cambridge University Press, 1982.

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

[Böd+09] Eckard Böde et al. "Compositional Dependability Evaluation for STATEMATE". In: *IEEE Trans. Software Eng.* 35.2 (2009), pp. 274–292.

[Boz+11] Marco Bozzano et al. "Safety, Dependability and Performance Analysis of Extended AADL Models". In: *Comput. J.* 54.5 (2011), pp. 754–775.

[Brá+12] Tomás Brázdil et al. "Verification of Open Interactive Markov Chains". In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*. 2012, pp. 474–485.

[Brá+13] Tomás Brázdil et al. "Continuous-time stochastic games with time-bounded reachability". In: *Inf. Comput.* 224 (2013), pp. 46–70.

[Bra+14] Bettina Braitling et al. "MeGARA: Menu-based Game Abstraction and Abstraction Refinement of Markov Automata". In: *Proceedings Twelfth International Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2014, Grenoble, France, 12-13 April 2014.* 2014, pp. 48–63.

[Bra+15] Bettina Braitling et al. "Abstraction-Based Computation of Reward Measures for Markov Automata". In: *Verification, Model Checking, and Abstract Interpretation - 16th International Conference, VMCAI 2015, Mumbai, India, January 12-14, 2015. Proceedings*. 2015, pp. 172–189.

[BS11] Peter Buchholz and Ingo Schulz. "Numerical analysis of continuous time Markov decision processes over finite horizons". In: *Computers & Operations Research* 38.3 (2011), pp. 651–659.

[Buc+11] Peter Buchholz, Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. "Model Checking Algorithms for CTMDPs". In: *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*. 2011, pp. 225–242.

[But+15] Yuliya Butkova, Hassan Hatefi, Holger Hermanns, and Jan Krcál. “Optimal Continuous Time Markov Decisions”. In: *Automated Technology for Verification and Analysis - 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings*. 2015, pp. 166–182.

[Cox55] D.R. Cox. “A use of complex probabilities in the theory of stochastic processes”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 51.02 (1955), pp. 313–319.

[DG07] Peng Dai and Judy Goldsmith. “Topological Value Iteration Algorithm for Markov Decision Processes”. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. 2007, pp. 1860–1865.

[DH11] Yuxin Deng and Matthew Hennessy. “On the Semantics of Markov Automata”. In: *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*. 2011, pp. 307–318.

[DH13] Yuxin Deng and Matthew Hennessy. “On the semantics of Markov automata”. In: *Inf. Comput.* 222 (2013), pp. 139–168.

[EHZ10a] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. “Concurrency and Composition in a Stochastic World”. In: *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*. 2010, pp. 21–39.

[EHZ10b] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. “On Probabilistic Automata in Continuous Time”. In: *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*. 2010, pp. 342–351.

[Eis+13] Christian Eisentraut, Holger Hermanns, Joost-Pieter Katoen, and Lijun Zhang. “A Semantics for Every GSPN”. In: *Application and Theory of Petri Nets and Concurrency - 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings*. 2013, pp. 90–109.

[FC08] James H Fowler and Nicholas A Christakis. “Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study”. In: *BMJ* 337 (2008). eprint: `http://www.bmj.com/content/337/bmj.a2338.full.pdf`.

[Fea+11] John Fearnley, Markus N. Rabe, Sven Schewe, and Lijun Zhang. “Efficient Approximation of Optimal Control for Continuous-Time Markov Games”. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2011, December 12-14, 2011, Mumbai, India*. 2011, pp. 399–410.

[Fea+16] John Fearnley, Markus N. Rabe, Sven Schewe, and Lijun Zhang. "Efficient approximation of optimal control for continuous-time Markov games". In: *Inf. Comput.* 247 (2016), pp. 106–129.

[For+11] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. "Automated Verification Techniques for Probabilistic Systems". In: *Formal Methods for Eternal Networked Software Systems - 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2011, Bertinoro, Italy, June 13-18, 2011. Advanced Lectures*. 2011, pp. 53–113.

[Fu14a] Hongfei Fu. "Maximal Cost-Bounded Reachability Probability on Continuous-Time Markov Decision Processes". In: *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*. 2014, pp. 73–87.

[Fu14b] Hongfei Fu. "Verifying Probabilistic Systems: New Algorithms and Complexity Results". PhD thesis. RWTH Aachen University, 2014.

[GBK16] Daniel Gburek, Christel Baier, and Sascha Klüppelholz. "Composition of Stochastic Transition Systems Based on Spans and Couplings". In: *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*. 2016, 102:1–102:15.

[GBL10] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. "Learning influence probabilities in social networks". In: *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*. 2010, pp. 241–250.

[GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. "The Google file system". In: *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003, SOSP 2003, Bolton Landing, NY, USA, October 19-22, 2003*. 2003, pp. 29–43.

[Gil76] Daniel T. Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions". In: *Journal of Computational Physics* 22.4 (1976), pp. 403–434.

[Gir12] Sergio Giro. "Efficient computation of exact solutions for quantitative model checking". In: *Proceedings 10th Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2012, Tallinn, Estonia, 31 March and 1 April 2012.* 2012, pp. 17–32.

[GSK05] Michele Garetto, Jingpu Shi, and Edward W. Knightly. “Modeling media access in embedded two-flow topologies of multi-hop wireless networks”. In: *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, MOBICOM 2005, Cologne, Germany, August 28 - September 2, 2005*. 2005, pp. 200–214.

[GSK06] Michele Garetto, Theodoros Salonidis, and Edward W. Knightly. “Modeling Per-Flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks”. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain*. 2006.

[Guc+12] Dennis Guck, Tingting Han, Joost-Pieter Katoen, and Martin R. Neuhäußer. “Quantitative Timed Analysis of Interactive Markov Chains”. In: *NASA Formal Methods - 4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012. Proceedings*. 2012, pp. 8–23.

[Guc+13] Dennis Guck et al. “Modelling, Reduction and Analysis of Markov Automata”. In: *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*. 2013, pp. 55–71.

[Guc+14a] Dennis Guck et al. “Analysis of Timed and Long-Run Objectives for Markov Automata”. In: *Logical Methods in Computer Science* 10.3 (2014).

[Guc+14b] Dennis Guck et al. “Modelling and Analysis of Markov Reward Automata”. In: *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*. 2014, pp. 168–184.

[Guc12] Dennis Guck. “Quantitative Analysis of Markov Automata”. MA thesis. RWTH Aachen University, June 2012.

[Hat+15] Hassan Hatefi et al. “Cost vs. Time in Stochastic Games and Markov Automata”. In: *Dependable Software Engineering: Theories, Tools, and Applications - First International Symposium, SETTA 2015, Nanjing, China, November 4-6, 2015, Proceedings*. 2015, pp. 19–34.

[Her+03] Holger Hermanns, Joost-Pieter Katoen, Joachim Meyer-Kayser, and Markus Siegle. “ETMCC: Model Checking Performability Properties of Markov Chains”. In: *2003 International Conference on Dependable Systems and Networks (DSN 2003), 22-25 June 2003, San Francisco, CA, USA, Proceedings*. 2003, p. 673.

[Her02] Holger Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*. Vol. 2428. Lecture Notes in Computer Science. Springer, 2002.

[HH12] Hassan Hatefi and Holger Hermanns. “Model Checking Algorithms for Markov Automata”. In: *ECEASST* 53 (2012).

[HH13a] Hassan Hatefi and Holger Hermanns. “Improving Time Bounded Reachability Computations in Interactive Markov Chains”. In: *Fundamentals of Software Engineering - 5th International Conference, FSEN 2013, Tehran, Iran, April 24-26, 2013, Revised Selected Papers*. 2013, pp. 250–266.

[HH13b] Holger Hermanns and Arnd Hartmanns. “An Internet Inspired Approach to Power Grid Stability”. In: *it - Information Technology* 55.2 (2013), pp. 45–51.

[HH15] Hassan Hatefi and Holger Hermanns. “Improving time bounded reachability computations in interactive Markov chains”. In: *Sci. Comput. Program.* 112 (2015), pp. 58–74.

[HHK00] Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. “On the Use of Model Checking Techniques for Dependability Evaluation”. In: *19th IEEE Symposium on Reliable Distributed Systems, SRDS’00, Nürnberg, Germany, October 16-18, 2000, Proceedings*. 2000, pp. 228–237.

[HK09] Holger Hermanns and Joost-Pieter Katoen. “The How and Why of Interactive Markov Chains”. In: *Formal Methods for Components and Objects - 8th International Symposium, FMCO 2009, Eindhoven, The Netherlands, November 4-6, 2009. Revised Selected Papers*. 2009, pp. 311–337.

[HKK13] Holger Hermanns, Jan Krcál, and Jan Kretínský. “Compositional Verification and Optimization of Interactive Markov Chains”. In: *CONCUR 2013 - Concurrency Theory - 24th International Conference, CONCUR 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*. 2013, pp. 364–379.

[Jen53] Arne Jensen. “Markoff chains as an aid in the study of Markoff processes”. In: *Scandinavian Actuarial Journal* 1953.sup1 (1953), pp. 87–91.

[Joh08] Sven Johr. “Model checking compositional Markov systems”. PhD thesis. Saarland University, 2008.

[Kar93] David R. Karger. “Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm”. In: *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.* 1993, pp. 21–30.

[Kat+09] Joost-Pieter Katoen et al. “The Ins and Outs of the Probabilistic Model Checker MRMC”. In: *QEST 2009, Sixth International Conference on the Quantitative Evaluation of Systems, Budapest, Hungary, 13-16 September 2009*. 2009, pp. 167–176.

[Kha80] Leonid G. Khachiyan. “Polynomial algorithms in linear programming”. In: *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* 20 (1980), pp. 51–68.

[Mil68] Bruce L. Miller. “Finite State Continuous Time Markov Decision Processes with a Finite Planning Horizon”. In: *SIAM Journal on Control* 6.2 (1968), pp. 266–280.

[MMS85] John F. Meyer, Ali Movaghar, and William H. Sanders. “Stochastic Activity Networks: Structure, Behavior, and Application”. In: *International Workshop on Timed Petri Nets, Torino, Italy, July 1-3, 1985*. 1985, pp. 106–115.

[Neu10] Martin R. Neuhäußer. “Model checking nondeterministic and randomly timed systems”. PhD thesis. RWTH Aachen University and University of Twente, 2010, pp. 1–244.

[Neu75] Marcel F. Neuts. “Probability distributions of phase type”. In: *Liber Amicorum Prof. Emeritus H. Florin, University of Louvain*. 1975, pp. 173–206.

[NSK09] Martin R. Neuhäußer, Mariëlle Stoelinga, and Joost-Pieter Katoen. “Delayed Nondeterminism in Continuous-Time Markov Decision Processes”. In: *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*. 2009, pp. 364–379.

[Pre+07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. New York, NY, USA: Cambridge University Press, 2007.

[Pul09] Reza Pulungan. “Reduction of Acyclic Phase-Type Representations”. PhD thesis. Saarland University, 2009.

[Put05] M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and statistics. Wiley-Interscience, 2005.

[QWP01] Qinru Qiu, Qing Wu, and Massoud Pedram. “Stochastic modeling of a power-managed system-construction and optimization”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20.10 (2001), pp. 1200–1217.

[RS11] Markus N. Rabe and Sven Schewe. “Finite optimal control for time-bounded reachability in CTMDPs and continuous-time Markov games”. In: *Acta Inf.* 48.5-6 (2011), pp. 291–315.

[RS13] Markus N. Rabe and Sven Schewe. “Optimal time-abstract schedulers for CTMDPs and continuous-time Markov games”. In: *Theor. Comput. Sci.* 467 (2013), pp. 53–67.

[RT87] Prabhakar Raghavan and Clark D. Tompson. “Randomized rounding: A technique for provably good algorithms and algorithmic proofs”. In: *Combinatorica* 7.4 (1987), pp. 365–374.

[Rud76] W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976.

[SB93] Frank A. Sonnenberg and J. Robert Beck. “Markov Models in Medical Decision Making: A Practical Guide”. In: *Med Decis Making* 13.4 (1993), pp. 322–338.

[Sch05] René L. Schilling. *Measures, Integrals and Martingales*. Cambridge Books Online. Cambridge University Press, 2005.

[Seg95] Roberto Segala. “Modeling and Verification of Randomized Distributed Real-Time Systems”. Available as Technical Report MIT/LCS/TR-676. PhD thesis. MIT EECS, June 1995.

[Sim+00] Tajana Simunic, Luca Benini, Peter Glynn, and Giovanni De Micheli. “Dynamic Power Management for Portable Systems”. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. MobiCom ’00. 2000, pp. 11–19.

[Std12] IEEE Std. “802.3-2012–IEEE Standard for Ethernet”. In: (Dec. 2012), pp. 1–3747.

[TBT06] Axel Thümmler, Peter Buchholz, and Miklós Telek. “A Novel Approach for Phase-Type Fitting with the EM Algorithm”. In: *IEEE Trans. Dependable Sec. Comput.* 3.3 (2006), pp. 245–258.

[Tim+12] Mark Timmer, Joost-Pieter Katoen, Jaco van de Pol, and Mariëlle Stoelinga. “Efficient Modelling and Generation of Markov Automata”. In: *Proc. of CONCUR*. Vol. 7454. LNCS. Springer, 2012, pp. 364–379.

[Tim11] Mark Timmer. “SCOOP: A Tool for SymboliC Optimisations of Probabilistic Processes”. In: *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*. 2011, pp. 149–150.

[Tim13] Mark Timmer. “Efficient Modelling, Generation and Analysis of Markov Automata”. PhD thesis. University Twente, 2013.

[TPS13] Mark Timmer, Jaco van de Pol, and Mariëlle Stoelinga. "Confluence Reduction for Markov Automata". In: *Formal Modeling and Analysis of Timed Systems - 11th International Conference, FORMATS 2013, Buenos Aires, Argentina, August 29-31, 2013. Proceedings*. 2013, pp. 243–257.

[WJ06] Nicolás Wolovick and Sven 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*. 2006, pp. 352–367.

[ZN10] Lijun Zhang and Martin R. Neuhäußer. "Model Checking Interactive Markov Chains". In: *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*. 2010, pp. 53–68.