Ca' Foscari University of Venice
Master's Degree Programme in Computer Science

Final Thesis

Approximate Persistent Stochastic Non-Interference

Academic Year 2018/2019

Supervisor
Ch. Prof.ssa Sabina Rossi

Graduand
Kotono Yoshida
Matricolation number 853696

Contents

Abstract

In this thesis a security property for stochastic, cooperating processes expressed as terms of the Performance Evaluation Process Algebra (PEPA) is studied. It is expressed as the notion of Persistent Stochastic Non-Interference (PSNI). This work consists in the attempt of relaxing the strict condition of PSNI by introducing a novel equivalence relation over PEPA components, named proportional bisimulation, which induces a proportionally lumpable partition on the state-space of the underlying Markov process.
Lumpability approach is a method to tackle the state space explosion problem by reducing the state space of a Markov chain. Equivalent states are aggregated into a unique partition, creating a new aggregated Markov chain that is smaller but its behaviour is the same as the original chain. However, the conditions for a partition on the original state space to be lumpable are quite strict. The introduction of proportional lumpability is then an attempt to relax the conditions in order to aggregate the states of the considered Markov chain. In line with this thinking, also the property PSNI can be, in some sense, relaxed by adopting the less strict form of the concept of lumpable bisimulation.
For this purpose, this thesis can be divided into two main sections: one is the study about the concept of proportional lumpability, the other is the application of proportional bisimulation based on proportional lumpability to the property PSNI.

Chapter 1

Introduction

1.1 Motivation

My thesis has started with the general idea of introducing a relaxed version of the security property called *Persistent Stochastic Non-Interference*, often shortened to *PSNI*. *PSNI* is a property which ensures the security of a system by looking at all the possible states reachable by the system taken into account and checking the *Stochastic Non-Interference* (*SNI*) property for each of them. *SNI* is based on the property called *Non-Interference*, which is an information flow security property which aims at protecting data from undesired accesses. *PSNI* is based on a structural operational semantics and a bisimulation based observation equivalence for the *PEPA* terms. The *PSNI* property can be widely exploited in order to protect the confidentiality of information by guaranteeing that high level, sensitive, information never flows to low level, unauthorized users. Indeed, attacks like the so-called covert channel can be possible even if computer security policy and/or cryptographic rules are used. Nevertheless, in real world situations it may be difficult to find a system that satisfies all the requirements of *PSNI*. Indeed, *PSNI* is based on the strong equivalence-like relation over *PEPA* components, called *lumpable bisimulation*, which requires strict conditions on the rates of components, in particular two components are considered lumpably bisimilar if the transition rates from these components to any equivalence class are the same. In order to propose a new version of *PSNI* property, first of all, the definition of a novel equivalence relation over PEPA components is necessary. The definition of this new equivalence relation naturally deals with the concept of *lumpability*, which also should be relaxed to produce relaxed partitions on the state space of the underlying Markov chain.

1.2 Performance Modelling

In this section, some preliminaries about the performance modelling theory, which consists in one of the central focuses of this work, will be presented. *Performance modelling* is a method used to model the dynamic behaviour of computer and communication systems for the purpose of identifying performance characteristics of the systems themselves, in such a way that optimization techniques can be exploited and applied to improve the systems' behaviour from the performance viewpoint. For example, one of the users' typical requirements consists in measurements of quantitative information such as *response time*, which is desired to be as small as possible; the *rate* at which something is processed, as known as *throughput*, should be as high as possible; *blocking probability*, also referred to as congestion, should be preferably zero, of course. These are called external measurements. In contrast to them, system managers may seek to optimize internal measurements like *resource utilization*, which is the usage of processing resource, should be reasonably high; *idle time* or non-productive time, should be as small as possible; *failure rate* which obviously is desired to be as low as possible, close to zero. The reason why users and/or system managers want to have the knowledge about all these measurements is that by having quantitative information of the entire system they can do the capacity planning, i.e., it is possible to answer the question *"how many clients can the existing server support and maintain reasonable response times?"* or alternatively, system managers can execute system configuration, by answering the following question: *"how many frequencies do I need to keep blocking probabilities low?"* and so on. It can be said that performance modelling allows one to represent a system through an abstract model which is used to encapsulate the characteristics of the entire system.

The history of performance modelling has already started in the early 20th century in order to emphasize the performance properties of communication systems. Nonetheless, performance modelling techniques for computer systems have been available only since the mid-1960s and they have been refined as the computers became more complicated and sophisticated.
Originally, *queueing theories* have been extensively applied to represent and analyse the very first computer system models. Indeed, it can be said that queueing networks are powerful and all-round tool for performance evaluation. A *queue* is based on the idea that users arrive to the queue and they wait some time, and afterwards they are processed so that they can leave the queue. The

basic queueing model is depicted in the figure here below, taken from [2].

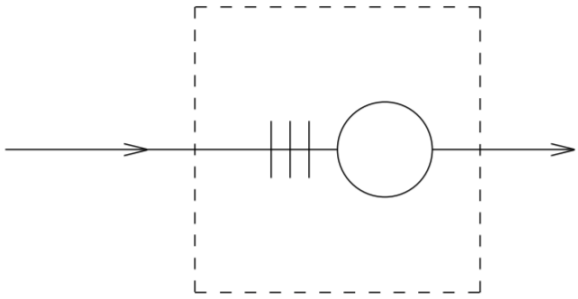

Figure 1.1: Basic queueing model

In 1909 a Danish engineer Agner Krarup Erlang published his first paper dealing with the queueing theory. Erlang was studying the problems about the congestion in telephone service for the Copenhagen Telephone Company: quantitative measurements such as waiting time and average number of waiting customers have been analysed in his paper. In [2] the basic queue model is characterized as follows:

- The arrival of users: usually users are assumed to arrive according to a Poisson distribution and they can arrive one by one, or alternatively, they can arrive in group.
- The behaviour of users: users can *decide* to wait until they are served, or leave the queue without waiting. Similar situations can be found in call centres: some customers are patient and will wait until an operator is available, other will hang up.
- The service time: the service (for example, the call with the operator of a call centre is the service) times are assumed to be independent and identically distributed.
- The service policy: there exist several disciplines of the service and according to it the queue serves the users. The queue usually serves in FIFO order, so first in first out, but it can also serve randomly, it depends.

- The service capacity: in call centre example, the capacity is the number of working operators. The server can be single or some quantity of them in such a way that the two users can be served at the same time.
- Waiting: there may be limitations on the number of waiting users in the queue. This concept is usually indicated as *buffer*, so only finite users can be buffered and served.

When no other techniques were available, even sophisticated system analysis were conducted by using queueing theories, but as computer systems have improved their performance, the use of classical method such as queueing theory has become unsatisfactory of course. Some of several challenges that a modelling theory should deal in order to model real computer systems' behaviour are listed up here below:

- *Time*: timing information such as network latency due to the physical distances is necessary;
- *Randomness and Probability*: the concept of random variables and notions about probability theory are quite important in order to deal with computer systems. For example, parts of a system like server or router may be down temporarily with some probability;
- *Scale*: there is the need to know the population size. It should be quantified in order to characterize correctly the workload. One should be able to correctly replicate the service to support all of the subscribers;
- *Percentages*: there is the need to know resource sharing percentages among the users of the modelled system, like the network contention, CPU loads.

In addition to all the above-mentioned features, the notions of *concurrency* and *parallelism* became gradually essential.

Petri nets are another type of well-established, both graphical and mathematical modelling tool available since the late 1960s, mainly used in order to model parallel and distributed systems. Petri nets consist in directed graphs in which the nodes are divided into two types, *places* and *transitions* that may be connected by directed arcs. Places represent the states in a system and transitions represent actions that may occur in a system. In this way, the behaviour of several systems can be represented in terms of system states and their changes. In the following lines, the formal definition of a Petri net presented in [46] is reported: a Petri net is a 5-tuple, $PN = (P, T, F, W, M_0)$ where:

- $P = \{p_1, p_2, ..., p_m\}$ is a finite set of places
- $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $W : F \to \{1, 2, 3, ...\}$ is a weight function,
- $M_0 : P \to \{0, 1, 2, 3, ...\}$ is the initial marking,
- $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

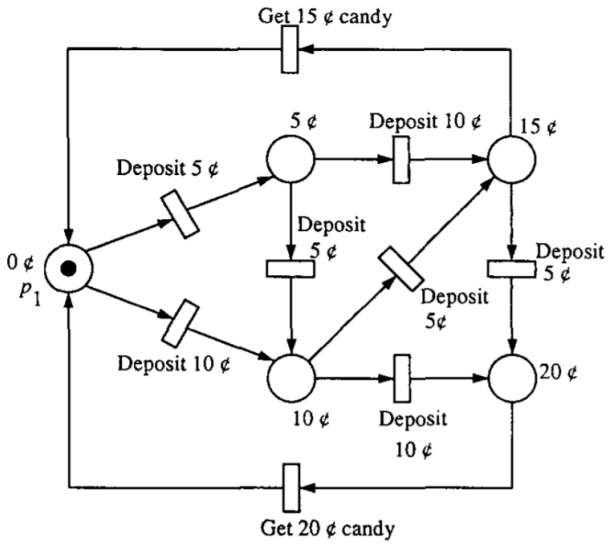

Figure 1.2: A Petri net (a state machine) representing the state diagram of a vending machine, where coin return transitions are omitted.

A Petri net structure $N = (P, T, F, W)$ without any specific initial marking is denoted by N. An example of the state diagram of a finite-state vending machine represented by Petri net is depicted in Figure 1.2, taken from [46]: the machine accepts pennies and sells either 15 cents candy bars or 20 cents candy bars. The buyer can deposit 5 cents for four times and get the 20 cents candy bars, or he can deposit 10 cents and then 5 cents so that he will get 15 cents candy bars, etc.

In [16] some advantages of Petri nets are described, among them, Petri nets can model graphically the system and so they allow an easy visualization of complex systems. Moreover, Petri nets allow hierarchy modelling, so that systems can be represented at various level of abstraction.
Both queueing networks and Petri nets are based on stochastic models. The main differences between them is that queueing theories offer compositionality but not formality; on the other hand, Petri nets offer formality but not compositionality. Consequently, a tool which is able to handle all of the above-mentioned attractive features in order to model a complex computer and communication systems of nowadays is needed.

1.3 Process Algebra and Bisimulation

In this section we are going to briefly introduce what is a process algebra and consequently we focus on PEPA language, by showing the concept of bisimulation which will be widely exploited during all this thesis.
In [7] a *process* is defined as "*behaviour of a system*", thus the process theory is the study of processes, namely the behaviour of a system. Process theory deals mainly with two tasks: *modelling* and *verification*. Through the techniques of modelling, as introduced in the previous section, processes are represented by mathematical expressions and formalisms; while verification consists in the activity of checking the correctness of processes, in order to prove, for example, whether the desired behaviour of the system is correctly obtained. Obviously, these two steps of modelling and verification are possible only if the precise semantics of the process theory is well defined. In [27], the author has defined process algebra as follows: "*process algebra constitutes a framework for formal reasoning about processes and data, with the emphasis on processes that are executed concurrently*". Moreover, the author of [4] has explained the reason why we use the term *algebra*: "*while elementary algebra is concerned with ma-*

nipulating numbers, a process algebra, or the synonymous term process calculus is concerned with the creation, life, and death of processes that carry out computations".
In this context, it can be stated that process algebra is part of mathematical theories that offer techniques which allow describing the dynamic behaviour of concurrent systems. As the name suggests, process algebras provide methods for high-level description of systems made of processes that interact with others within the same environment: in particular, algebraic laws that permit the description and formal reasonings on processes are available. In case the models are characterized by quantitative information, the process algebra assumes stochasticity and thus it will be called *stochastic process algebra*. The foundations of process algebra are partly rooted in Petri nets, briefly presented in the previous section, and automata theory. Then its advent was greatly marked by the introduction of the *CCS* language in the seminal monograph *A Calculus of Communicating Systems* by Milner, published in 1980. Later in 1984, Brookes, Hoare and Roscoe elaborated the *CSP* in the paper *A theory of communicating sequential processes*; at the same time, Bergstra and Klop published their paper *Process algebra for synchronous communication* in which they presented *ACP*. The most well-known examples of process algebras are listed up here below:

- *Calculus of Communicating Systems* (*CCS*)
  As it has been stated before, *CCS* has been introduced by Robert Milner in 1980, and it consists in a process algebra as a model of concurrent system. In *CCS* there exist two main components: *agents*, they are the active components within the system; *actions*, they are performed by agents and determine their behaviour.
  CCS has been thought as a tool for describing the behaviour of systems, consisting of subsystems which communicate concurrently. The key idea exploited by Milner is the following: the concurrent behaviour can be considered as the one seen by an external observer. This idea gives rise to the concept of observation equivalence, which is conceptually really similar to the bisimilarity, which will be widely used along this thesis.

- *Communicating Sequential Processes* (*CSP*)
  CSP has been introduced in 1984 by Brookes, Hoare, Roscoe and of course it is part of process algebra too. As opposed to *CCS*, *CSP* seems to be more programming language-like expression, so that it inspired the programming language *occam*, used for parallel programming nowadays.

Indeed, *CSP* was born as a theoretical version of a practical language for concurrency, [6]. The common element between *CSP* and *CCS* is that both of them are founded on the concept of process and process within a system. However, *CSP* provides two types of operator choice, internal and external. In *CCS* there is no such a differentiation.

- *Algebra of Communicating Processes* (*ACP*)
  As the name suggests, *ACP* is part of the family of process algebras, but it is focused on the algebra of processes. In particular, while in *CCS* the semantics is operational and in terms of labelled transition systems, in *ACP* semantics is algebraic.

In particular, *CCS* have been widely used in order to verify whether the system is behaving correctly and this type of modelling is called *qualitative modelling*. Look at the two pictures here below: the following two networks are equivalent in the sense that they exhibit the same connectivity property with respect to the receivers. Note that, these two pictures are taken from materials of *Formal Methods for System Verification* course of *Ca' Foscari University of Venice*, by prof. Sabina Rossi.

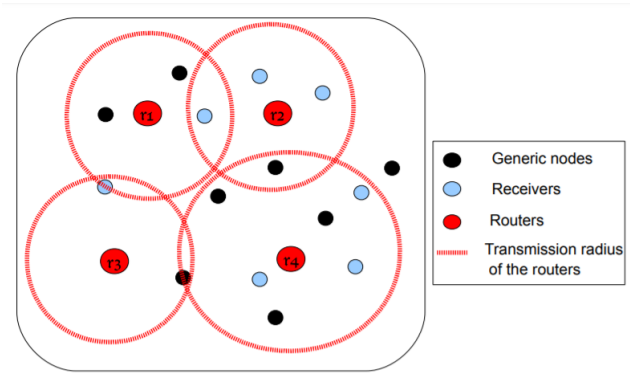

Figure 1.3: Connectivity property example 1

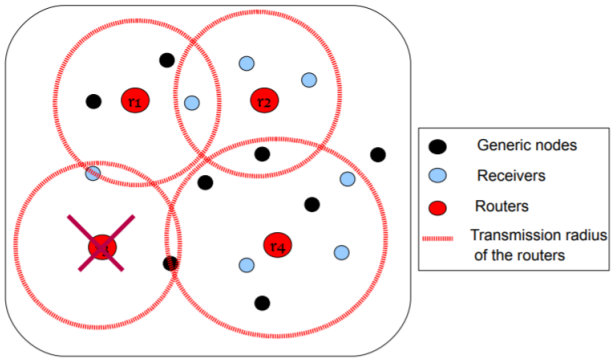

Figure 1.4: Connectivity property example 2

Notice that in pure process algebras all actions are considered to be instantaneous, so that the timing information is not incorporated within it. There exists a variation of *CCS*, called *SCCS* which is the synchronous version of *CCS*: in *SCCS* implicit global clock is considered to be working in such a way that each action occurs at each clock tick.

The authors of [20] state that the *Performance Evaluation Process Algebra* (*PEPA*) which will be presented in its details in Chapter 3, is an algebraic description technique based on a classical process algebra, as the ones introduced above. In contrast to the classical process algebras such as *CCS*, in PEPA timing information is not abstracted away. Indeed, in *CCS*, which is a pure process algebra, actions are assumed to be instantaneous, there is no concept of duration of each activity. If an exponentially distributed random variable is used to specify the duration of each action the process algebra may be used to represent a Markov process. This approach is taken in PEPA and the other recently published stochastic process algebras [8, 32].

There can exist several motivations of the use of stochastic process algebra such as PEPA for performance modelling, among them:

- *Integrating Performance Analysis into System Design*: it is important to consider the timing information of performance aspects of a system taken into account;
- *Representing Systems as Models*: queueing theories offer too restricted

expressiveness, in particular queueing networks are not suitable for modelling computer and telecommunication systems;

- *Model Tractability*: modelling real world systems may engage in the creation of enormous huge model with a lot of states within it. This has led to considerable interest in model simplification and aggregation techniques and scalability techniques in order to represent the system at different abstraction levels. Indeed, process algebras include mechanisms for composition and abstraction. These mechanisms facilitate the systematic development of large models with hierarchical structure.

Furthermore, it can be said that process algebra style of system representation is close to the way that designers describe the systems. In this way, performance analysis can be integrated into design methodologies, consequently by allowing to perform both qualitative and quantitative modelling using the same system description. Finally, a process algebra description represents a system as a collection of active agents who cooperate to achieve some behaviour of the entire system. This cooperation paradigm is really useful for modelling many modern computer systems which consist in both autonomous components and cooperating ones. A stochastic process algebra such as *PEPA* has been designed with the purpose of filling the gaps of previous performance modelling paradigms, with other important and useful features such as *formality*, which helps in giving a precise meaning to all terms in the language; *abstraction*, the ability to build up complex models even from simple bricks, so that it is possible to scale the abstraction level adapting it case by case, and the most importantly *compositionality*, the ability to model a system as the interaction of subsystems.

Another important characteristic of the process algebra PEPA, which cannot be ignored, consists in the concept of *bisimulation*. As already expressed in the previous section, this thesis deals with the notion of lumpable bisimulation over PEPA components, since our main focus is the property *PSNI* which is founded on the above-mentioned equivalence relation. PEPA is a process algebra which has been ideated for a specific purpose, that is performance evaluation of modelled system. For this reason, sometimes there can exist the need to establish the *performance equivalence* between two systems: indeed evaluation often deals with comparisons with other systems in order to decide which is performing better. Performance equivalence, in simple words is going to establish if the two considered systems are able to perform same set of

actions within the same period of time, and in PEPA this is captured by the bisimulation relation. In particular, the notion of bisimulation is strongly connected with the observation from external point of view, that is, two systems are considered as one whether their behaviour from external viewpoint appears to be the same. However, we should specify that there are mainly two kind of bisimulation, a weaker one and a stronger one; in particular, the weaker form of bisimulation allows private actions within an agent, so that these actions cannot be captured and observed by an external viewpoint.
A bisimulation relation allows one to create equivalence classes over the set of agents composing the considered system S and this partition results to be useful in order to make the system smaller, so that two equivalent agents are considered as one. In this thesis, an alternative notion of equivalence over PEPA components, in the bisimulation style, will be widely used: strong equivalence and in particular, its variant which is lumpable bisimulation. The notion of strong equivalence is founded on the idea of the probabilistic bisimulation relation, explained in [35] as an equivalence relation such that, for any two elements belonging to the same equivalence class, some probabilistic quantity of each of the considered elements performing the same fixed action is the same.
The equivalence relation lumpable bisimulation will be exploited in order to formally define the property *PSNI* and for this reason, we should first of all introduce it before studying *PSNI* in its details. The notions of strong equivalence and lumpable bisimulation are presented in Chapter 4. Notice that, strong equivalence and lumpable bisimulation, as the latter one suggests, they both are based on the concept of lumpability of the underlying Markov chains. In simple terms, a process should be ordinarily or strongly lumpable with respect to the partition generated by some equivalence relation, in order to be ensured that Markov property is preserved. Details about Markov chains and lumpability will be discussed in Chapter 2.

1.4 Stochastic Non-Interference

As briefly explained in the abstract and in the previous section, the main focus of this thesis is the *PSNI* property. *PSNI* is the acronym of *Persistent Stochastic Non-Interference* and we can see that two adjectives, persistent and stochastic are characterizing the well known *non-interference* concept: indeed, non-interference is one of the most widespread security policy, firstly introduced by Goguen and Meseguer in 1982, which aims to guarantee the safety

of a computer system by requiring that whenever some user is working with highly sensitive data (*e.g.*, the login with username and password to some site's mypage), the system should behave exactly as it is not processing any sensitive data, in order to protect them from uncleared possibly malicious user. The study of non-interference policy has begun in the beginning of 1980s with the purpose of clarifying the cause of *covert channel attacks*: indeed in these years only few theoretical security property were available. With the term covert channel attacks we indicate those cyberattacks in which there exists a transfer of information between processes not allowed to communicate each other. Despite the existence of computer security polices like access control policies, the covert channel attacks make possible the communication between agents which are not supposed to interact. In order to capture the problems of those attacked computer systems, non-interference policy has been ideated. The non-interference property, whenever characterized by the adjective *stochastic* will assume stochasticity, indeed the property *Stochastic Non-Interference* is a non-interference property specified as terms of the quantitative stochastic process algebra PEPA. We can say that *Stochastic Non-Interference* is a quantitative-stochastic extension of the well known property *Non-Interference*.
In this thesis, the persistent version of *Stochastic Non-Interference* namely *Persistent Stochastic Non-Interference* is studied, since we naturally are interested in the fact that this security property is maintained also for every process reachable by the system. The property *PSNI* seems to be really endearing, in the sense that, if a system is made of processes which are all satisfying the property *PSNI* we are ensured that no information flow will occur, avoiding serious attacks like covert channel attacks. However, we must say that the definition of *PSNI* is founded on lumpable bisimulation, meaning that it requires that the following systems are indistinguishable from external observer: the system which deals with some highly sensitive data and the same system but prevented from performing any highly sensitive activities. This kind of requirement is quite rarely satisfied in real world systems, indeed it is hard that the behaviour of two different systems appears perfectly the same. For this reason, in Chapter 7 a new moderate version of *PSNI* will be presented, which will be naturally founded on the relaxed version of lumpable bisimulation, which will be presented in Chapter 5.

1.5 Contribution

Now that the main concepts and preliminaries about the work of this thesis have been presented, my personal contribution now can be listed up as follows:

- Definition of a new characterization of lumpability concept, called *proportional lumpability* and its study: since strong lumpability is too much restrictive, a new relaxed and coarser version of it is introduced. We will see that proportional lumpability is characterized by several interesting properties that allow one to derive exact performance indices for the original process;
- Definition of a new equivalence relation over PEPA components, called *proportional bisimulation*: as like as the lumpablity case, also lumpable bisimulation, the equivalence relation over PEPA components, results to be too much strict so that, the corresponding relaxed version which induces a grosser form of aggregation of components is presented;
- Definition of the new version of *PSNI*, by using the newly introduced notion of bisimulation: since *PSNI* is founded on the notion of lumpable bisimulation, we can now exploit the novel equivalence relation proportional bisimulation in order to define a new relaxed version of *PSNI*, which has been named as *Approximate-PSNI*, shortened as *A-PSNI*;
- Definition of the decision algorithm for *A-PSNI*, which is the new version of the one presented for *PSNI*: this consists in the extension of the algorithm proposed by the authors of [37, 36].

This thesis is structured as follows: in Chapter 1 some preliminaries about performance modelling and process algebra are presented. In Chapter 2 theoretical background on Continuous Time Markov Chains and concept of lumpability are recalled; moreover, *proportional lumpability* is introduced and studied in its details. In Chapter 3 the *Performance Evaluation Process Algebra* (*PEPA*) is reviewed, by reporting all the semantics and rules which allows to compute performance measures of modeled systems. In Chapter 4 the well-known notion of equivalence, *strong equivalence*, and its variant *lumpable bisimulation* are presented. In Chapter 5 the novel equivalence relation, *proportional bisimulation* is introduced by showing its properties. In Chapter 6 the study and analysis on security property *PSNI* is reported, and in Chapter 7 the new

PSNI applying the novel notion of equivalence, the *A-PSNI*, is analysed in its details. In Chapter 8 a decision algorithm for *A-PSNI* is studied.
Finally, Chapter 9 concludes the thesis.

Chapter 2

Continuous Time Markov Chain

In the field of stochastic processes study, *Markov chains*, named after the Russian mathematician Andrey Markov, are a considerably powerful mathematical tool used in order to make predictions about the future of the process based exclusively on its present state. In other words, the key idea on which Markov chains are founded is the *memorylessness* property. Markov chains are widely used to model several real-world situations, from weather forecasting problem to search engine algorithm *PageRank* equation problem. Furthermore, also in economics and finance field the Markov processes are used in a wide variety of situations.

In this chapter we introduce some notions of the theory of Markov processes which will be required in the rest of the thesis. We mainly focus on Continuous Time Markov processes with a discrete space, the so-called *CTMC*s, but the presented arguments can be formulated also for Discrete Time Markov Chains (DTMCs). This chapter is structured as follows. Section 1 introduces the fundamental notions and notations of Markov processes. Section 2 deals with the study of the notion of lumpability. In Section 3 the concept of perturbation on Markov chain is studied, by introducing new kind of lumpability.

2.1 Preliminaries on Markov processes

In this section some theoretical background about Markov processes will be presented, by following the steps performed by the authors of [43].

Henceforth, we will indicate by $X(t)$ a stochastic process characterized by a discrete (countable) state space $\mathcal{S}$ for $t \in \mathbb{R}^+$. $X(t)$ is said to be a *stationary* stochastic process if $(X(t_1), X(t_2), ..., X(t_n))$ has the same distribution as the process
$(X(t_1 + \tau), X(t_2 + \tau), ..., X(t_n + \tau))$ for all $t_1, t_2, ..., t_n, \tau \in \mathbb{R}^+$.
Moreover, the stochastic process $X(t)$ results to be a *Markov* process if the behaviour of future states does not matter with the its past, the future depends only upon the present state, not on the sequence of past events. More formally, the joint distribution of $(X(t_1), X(t_2), ..., X(t_n))$ for $t_1 < t_2 < \cdots < t_{n+1}$ is such that

$$P(X(t_{n+1}) = i_{n+1} \mid X(t_1) = i_1, X(t_2) = i_2, ..., X(t_n) = i_n) =$$
$$P(X(t_{n+1}) = i_{n+1} \mid X(t_n) = i_n).$$

A Markov process $X(t)$ is:

- *Time homogeneous* if

$$P(X(t + \tau) = x_i \mid X(t) = x_j) = P(X(t' + \tau) = x_i \mid X(t') = x_j)$$

  In other words, the behaviour of the system does not depend on when it is observed, the transitions between states are independent of the time at which the transitions occur;

- *Irreducible* if from all states belonging to the state space $\mathcal{S}$ it is possible to reach every other state; it means that there exists a path between any pair of nodes in the directed graph which nodes are the states in the system, in other words, the graph is strongly connected.

A state x_i in a Markov process is called:

- *Recurrent* or *persistent* if the probability that the process will sooner or later return to x_i is 1. Otherwise, the state is called *transient*. A recurrent state x_i is said to be *positive-recurrent* if the required number of steps in order to go back to it is less than infinity.

A Markov process is *ergodic* if it is *irreducible* and all its states are *positive-recurrent*.

For every time-homogeneous, finite, irreducible Markov process, if a process satisfies all the above assumptions possesses a *steady-state distribution* (or *equilibrium*) that is the *unique* collection of positive real numbers $\pi(s)$ with $s \in \mathcal{S}$ such that:

$$\lim_{t\to\infty} P(X(t) = s \mid X(0) = s') = \pi(s)$$

where $\pi(s) \in \mathbb{R}^+$.

Steady-state probability distribution is the probability distribution of $X(t)$ as the system settles into a regular pattern of behaviour.

The transition rate between two states i and j is defined as follows:

$$q_{ij} = \lim_{\tau\to 0} \frac{Pr(X(t+\tau)) = x_j \mid X(t) = x_i}{\tau}$$

with $i \neq j$.

Let $X(t)$ be a Markov process with the state space $\mathcal{S}$, with $|\mathcal{S}|= N$; the *infinitesimal generator matrix* $\boldsymbol{Q}$ is the $N \times N$ matrix in which each off-diagonal entry q_{ij} is the transition rate of proceeding from state x_i to x_j, while the diagonal elements are formed as the negative sum of the non-diagonal elements of each row, i.e., $q_{ii} = -\sum_{h\in\mathcal{S},h\neq i} q_{ih}$.

In steady state, $\pi(i)$ is the proportion of time that the process spends in state x_i. We call *probability flux* from state x_i to state x_j the probability that a transition will occur from state x_i to state x_j: it is the probability $\pi(i)$ multiplied by the transition rate q_{ij}. In steady-state, the equilibrium is maintained in the following way: for any state the total probability flux out is equal to the total probability flux into the state. Consider the next equation:

$$\pi(i) \sum_{x_j\in\mathcal{S},j\neq i} q_{ij} = \sum_{x_j\in\mathcal{S},j\neq i} \pi(j) q_{ij}$$

We can reformulate the above equation as follows:

$$\boldsymbol{\pi Q} = \mathbf{0}$$

recalling that the diagonal elements of the infinitesimal generator matrix $\boldsymbol{Q}$ are $q_{ii} = -\sum_{x_j\in\mathcal{S},j\neq i} q_{ij}$. Therefore, we can rearrange the flux balance equation as

$\sum_{x_j \in \mathcal{S}} \pi(j) q_{ji} = 0$. Finally, $\pi(i)$ can be expressed as a row vector π, and consequently we can write as a matrix equation $\pi Q = 0$, the so-called *global balance equations (GBE)*. Any non-trivial solution of the GBE differs by a constant but only one satisfies the normalising condition $\sum_{k \in \mathcal{S}} \pi(k) = 1$, this is due to the fact that π is a probability distribution.

2.2 Aggregation and Lumpability

In this section the notion of *lumpability* is presented: it provides a model reduction technique which allows to generate an aggregated Markov chain, smaller with respect to the original one. The advantage consists not only in the dimension reduction but in particular on the possibility to compute exact results for the original chain from the aggregated one.

2.2.1 Background

The concept of *lumpability* provides a method for model simplification which helps in creating an aggregated Markov process that is smaller than the original one but behaves as the original process. Indeed, if the Markov process $X(t)$ is characterized by N states, its infinitesimal generator matrix Q will surely have dimension $N \times N$. However, this size often results too big to fit into the memory. This problem is known as *state space explosion problem*, indeed the author of [53] has stated that this problem makes the general algorithms for the performance and reliability analysis time and space consuming.
In order to tackle the problem, an approach is to *aggregate "similar"* states and establish a partition of the state space, in order to reduce their number. The resulting chain is therefore smaller with respect to the original one, but it can be efficiently exploited for the purpose of determining several measurement results for the original chain, without committing error. As the authors of [13] suggest, lumpability allows the definition of an aggregation of the Markov chain which consequently allows the exact determination of stationary results for the original chain. In this way, the heavy computation of several measures on the original chain can be avoided.
Before aggregating states, we should define an *equivalence relation* over the state space of the Markov process taken into account. In general, when a *CTMC* is aggregated, the aggregated stochastic process does not necessarily maintain the Markov property. The property is preserved in case the parti-

tion satisfies the so-called *strong lumpability condition*: indeed, the authors of [41] have proved that for an equivalence relation $\sim$ over the state space of a Markov process $X(t)$, the aggregated process is a Markov process for every initial distribution if and only if $\sim$ is a strong lumpability for $X(t)$. First of all, we should properly define what is an equivalence relation.

Definition 2.2.1. (Equivalence Relation) *Let $\sim$ be a binary relation on a set of states $\mathcal{S}, \sim \subseteq \mathcal{S} \times \mathcal{S}$. $\sim$ is said to be an* equivalence relation *if and only if it is reflexive, symmetric and transitive. That is, for all s, s' and $s'' \in \mathcal{S}$:*

- *reflexivity*: $s \sim s$
- *symmetry*: $s \sim s'$ if and only if $s' \sim s$
- *transitivity*: if $s \sim s'$ and $s' \sim s''$ then $s \sim s''$

2.2.2 Strong Lumpability

Consider the equivalence relation $\sim$ over the state space of a Continuous Time Markov Chain $X(t)$. Its original state space is defined as follows: $\mathcal{S} = \{0, 1, ..., N\}$. The *equivalence class* of s under $\sim$, denoted $[s]_\sim$, is defined as $[s]_\sim = \{s' \in \mathcal{S} \mid s \sim s'\}$. The set of all equivalence classes is denoted by $\mathcal{S}/\sim = \{[s_0]_\sim, [s_1]_\sim, ..., [s_m]_\sim\}$, with $m \leq N$, hopefully $m \ll N$. Henceforth, the following notations will be used in order to express the aggregated transition rates from a state within an equivalence class to some other equivalence class and vice versa:

$$q_{i[k]} = \sum_{j \in [k]_\sim} q_{ij}$$

$$q_{[k]i} = \sum_{j \in [k]_\sim} q_{ji}$$

Hereafter, the simple notation $[s_0]$ will be used in order to denote the equivalence class $[s_0]_\sim$ relative to the equivalence relation $\sim$.

The concept of *strong lumpability* has been introduced in [41] and further studied in [13, 54]. In the following lines the formal definition of strong lumpability is reported:

Definition 2.2.2. (Strong Lumpability) *Let $X(t)$ be a Continuous Time Markov Chain characterized by its own state space $\mathcal{S} = \{0, 1, ..., N\}$ and $\sim$ be an equivalence relation over the state space $\mathcal{S}$ of the considered chain. It can be said that $X(t)$ is* strongly lumpable *with respect to $\sim$ if for any equivalent class in $\mathcal{S}/\sim$, $[k] \neq [l]$ and $i, j \in [l]$, it holds that $q_{i[k]} = q_{j[k]}$.*

Therefore, we can say that a strongly lumpable partition exists whether there exists an equivalence relation over the state space of a Markov process, such that for any two states within an equivalence class their aggregated transition rates to any other class are coincident. Note that every Markov process is trivially strongly lumpable with respect to the identity relation. An example of a strongly lumpable partition is given in the following picture.

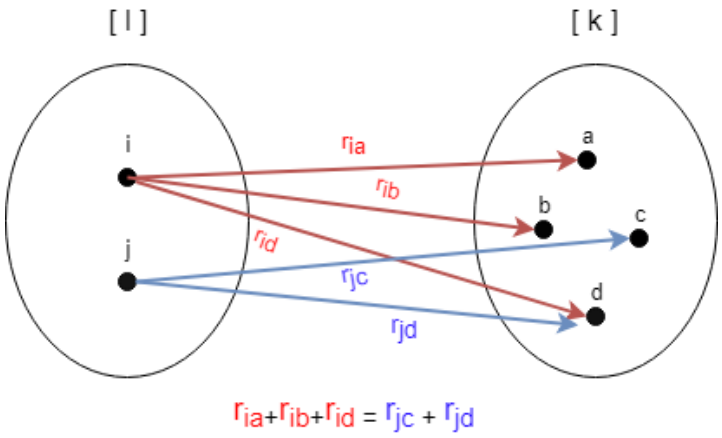

Figure 2.1: Example of strongly lumpable partition

The authors of [43] states that for an equivalence relation $\sim$ over the state space of a Markov process $X(t)$, the aggregated process is a Markov process for every initial distribution if and only if $\sim$ is a strong lumpability for $X(t)$.

Now, consider an equivalence relation $\sim$ over the state space of a Continuous Time Markov Chain $X(t)$. The aggregated CTMC according to the equivalence relation $\sim$, is denoted by $\widetilde{X}(t)$. In case the relation $\sim$ is a strong lumpability then $\widetilde{\boldsymbol{Q}} = (\widetilde{q}_{[i][j]})_{[i],[j]\in\mathcal{S}/\sim}$ will be the infinitesimal generator of $\widetilde{X}(t)$, and its definition proposed in [43] is reported here below:

Proposition 1. (Aggregated process) *Let $X(t)$ be a Continuous Time Markov Chain and $\sim$ be an equivalence relation over the state space of the considered Markov chain $X(t)$. The next two statements are considered equivalent:*

- *$\sim$ is a strong lumpability for $X(t)$;*
- *$\widetilde{X}(t)$ is a Markov process.*

Furthermore, if $\sim$ is a strong lumpability for the chain $X(t)$ then for all $[i], [j] \in \mathcal{S}/\sim$, $\widetilde{q}_{[i][j]} = q_{i[j]}$ in which $\widetilde{Q}$ results to be the infinitesimal generator matrix of $\widetilde{X}(t)$.

In the following lines, two examples of strongly lumpable Continuous Markov chains are presented.

Example 1 The Figure 2.2 is depicting a CTMC, with rates $\rho \neq \nu$.
$\mathcal{S}$ is its state space, $\mathcal{S} = \{1, 2, 3, 4\}$ and $\sim$ is the equivalence relation which is going to aggregate the states in the following way: $1 \sim 3$ and $2 \sim 4$, therefore it will induce the partition $\mathcal{S}/\sim = \{[s_{1,3}], [s_{2,4}]\}$. We can see that $\sim$ is a strong lumpability for $X(t)$.

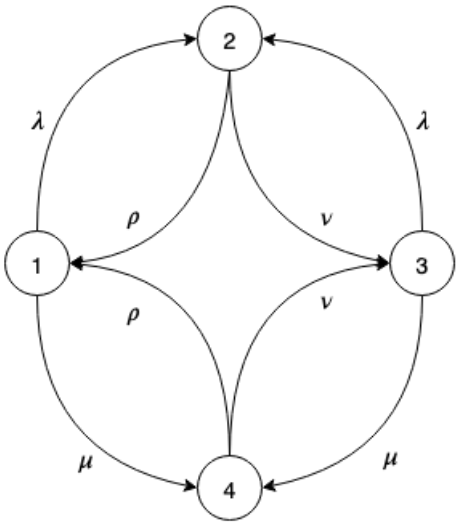

Figure 2.2: Example of strongly lumpable CTMC

Example 2 The Figure 2.3 is depicting a CTMC, characterized by its state space $\mathcal{S} = \{i_1, i_2, j_1, j_2, j_3\}$. $\sim$ is the considered equivalence such that $i_1 \sim i_2$, $j_1 \sim j_2$ and $j_2 \sim j_3$. It will induce the partition of the state space $\mathcal{S}/\sim = \{[i], [j]\}$ where $[i] = \{i_1, i_2\}$ and $[j] = \{j_1, j_2, j_3\}$.

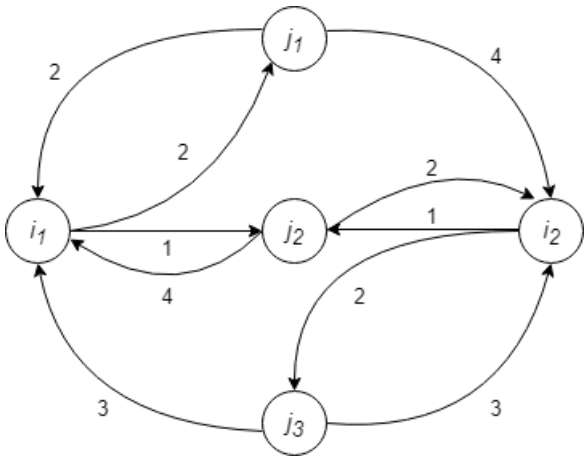

Figure 2.3: Example of strongly lumpable CTMC

2.2.3 Aggregated process

In this section the details about aggregated process are studied.
Also in this context, we should consider $X(t)$, a Continuous Time Markov Chain CTMC with discrete state space $\mathcal{S}$ for $t \in \mathbb{R}^+$. Its steady state distribution vector is indicated as $\boldsymbol{\pi}$. The members of the vector π must sum to unity and satisfy the system of global balance equation $\boldsymbol{\pi Q} = \mathbf{0}$. Nevertheless, the process of finding the solution of the latter system may engage in laborious computations. Indeed, the CTMC underlying the model of a real systems can possess great number of states. In order to simplify the computation we may aggregate the states of the CTMC taken into account.
Formally, let us consider an equivalence relation $\sim$ over the state space $\mathcal{S}$ of $X(t)$ defined in previous lines. The aggregated chain can be defined as follows: the state space of the aggregated chain consists in the set of the equivalence classes denoted as $\mathcal{S}/\sim$ and its infinitesimal generator matrix $\widetilde{\boldsymbol{Q}}$ can be built from the next equation:

For any $[l], [k] \in \mathcal{S}/\sim$,

$$\widetilde{q}_{[l][k]} = \frac{\sum_{i\in[l]} \pi(i) q_{i[k]}}{\sum_{i\in[l]} \pi(i)}$$

Subsequently, the next proposition states that the steady-state probability of each macro-state of the aggregated chain is the sum of the steady-state probabilities of the states in the original chain forming it.

Proposition 2. *Consider an ergodic Continuous Time Markov Chain $X(t)$ characterized by state space $\mathcal{S}$. Additionally, let $\sim$ be an an equivalence relation over $\mathcal{S}$. We denote with $\widetilde{X}(t)$ the aggregated process with respect to $\sim$. $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\pi}}$ are the steady-state distribution of $X(t)$ and $\widetilde{X}(t)$, respectively. For all $[s] \in \mathcal{S}/\sim$,*

$$\widetilde{\pi}([s]) = \sum_{s\in[s]} \pi(s).$$

Proof. For all $[s] \in \mathcal{S}/\sim$, we can write down the global balance equation as follows:

$$\widetilde{\mu}([s]) \sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \widetilde{q}_{[s][s']} = \sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \widetilde{\mu}([s'])\widetilde{q}_{[s'][s]}. \tag{2.1}$$

If we substitute $\widetilde{q}$ and $\widetilde{\pi}$ as defined previously, the left-hand side of the Equation 2.1 evolves as follows:

$$\begin{aligned}
&(\sum_{s\in[s]} \pi(s)) \sum_{[s']\in\mathcal{S}/\sim} \frac{\sum_{s\in[s]} \pi(s) \sum_{s'\in[s']} q_{ss'}}{\sum_{s\in[s]} \pi(s)} = \\
&(\sum_{s\in[s]} \pi(s)) \frac{\sum_{s\in[s]} \pi(s) \sum_{[s']\in\mathcal{S}/\sim} \sum_{\substack{s'\in[s'] \\ s'\neq s}} q_{ss'}}{\sum_{s\in[s]} \pi(s)} = \\
&\sum_{s\in[s]} \pi(s) \sum_{[s']\in\mathcal{S}/\sim} \sum_{s'\in[s']} q_{ss'} = \\
&\sum_{s\in[s]} \pi(s) \sum_{s'\in\mathcal{S}, s'\notin[s]} q_{ss'}.
\end{aligned}$$

Equivalently, it is possible to write down the right-hand side of Equation 2.1 as follows:

$$\sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \left(\sum_{s'\in[s']} \pi(s') \right) \frac{\sum_{s'\in[s']} \pi(s') \sum_{s\in[s]} q_{s's}}{\sum_{s'\in[s']} \pi(s')} =$$

$$\sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \sum_{s'\in[s']} \pi(s') \sum_{s\in[s]} q_{s's} =$$

$$\sum_{s\in[s]} \sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \sum_{s'\in[s']} \pi(s') q_{s's} =$$

$$\sum_{s\in[s]} \sum_{s'\in\mathcal{S}, s'\notin[s]} \pi(s') q_{s's}.$$

Therefore, it can be verified that for any $S \subseteq \mathcal{S}$

$$\sum_{s\in[s]} \pi(s) \sum_{s'\in\mathcal{S}, s'\notin[s]} q_{ss'} = \sum_{s\in[s]} \sum_{s'\in\mathcal{S}, s'\notin[s]} \pi(s') q_{s's}.$$

The Proposition 2 is a fundamental achievement because thanks to this result one can execute the analysis on several measurements on the aggregated chain instead of the original chain, characterized by a large amount of states, by reducing the computational time by far.

2.3 Perturbed Markov Chains

So far, theories and results on Markov chains without considering any changes in transition probabilities have been considered and analysed. However, if small differences on these transitions are allowed, also an *approximate* form of Markov chain can be considered, dealing thus with *perturbation theory*. More specifically, we can try to understand how several properties on the original Markov chain can persist under some form of perturbations. Therefore, with the term perturbation methods concerning the finding of approximate solution to a problem will be indicated.
In this context, the determination of *bounds* for interested measures on Markov chains is important. While in previous section Proposition 2 has been shown in order to prove that from the aggregated chain results for the overall chain can be computed without errors, if we allow some perturbation on the aggregated

chain, the bounds must be clearly determined in order to find errors made when computing the measures for the original chain from the aggregated one. The authors of [15] have compared several perturbation bounds for the stationary distribution of a Markov chain. The change between two steady-state distributions have been expressed as follows. Let the perturbed $\boldsymbol{Q}$ be the infinitesimal generator matrix, indicated as $\widetilde{\boldsymbol{Q}}$, then let $\widetilde{\boldsymbol{\pi}}$ be the steady-state distribution vector of $\widetilde{\boldsymbol{Q}}$. The change between these two vectors $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\pi}}$ can be easily expressed in terms of the change $E \equiv Q - \widetilde{Q}$ as follows:
For suitable norms and for various different condition numbers κ,

$$\|\pi - \widetilde{\pi}\| \leq \kappa \|E\|$$

There exist several condition numbers κ presented in [15] and among them those who do not depend on the steady-state probability vector $\boldsymbol{\pi}$ can be interesting to consider. Notice that these numbers are in terms of the group inverse of A, defined as $A = I - Q$; the group inverse of A is denoted by $A^{\#}$ satisfying $AA^{\#}A = A, A^{\#}AA^{\#} = A^{\#}, AA^{\#} = A^{\#}A$. Moreover, κ_5 and κ_6 are in terms of the *ergodicity coefficient* τ; the ergodicity coefficient $\tau_1(B)$ of a matrix B with equal row sums b is defined in [49] as follows: $\tau_1(B) \equiv \sup_{||v||_1=1, v^T e=0} ||v^T B||_1$.

The several types of conditions number are listed up here below, followed by references:

- $\kappa_2 = ||A^{\#}||_\infty$, Meyer [40]
- $\kappa_3 = \frac{max_j(a^{\#}_{jj} - min_i\ a^{\#}_{ij})}{2}$, Haviv and Van der Heyden [33]
- $\kappa_4 = max_{i,j}\ |a^{\#}_{ij}|$, Funderlic and Meyer [29]
- $\kappa_5 = \frac{1}{1-\tau_1(Q)}$, Seneta [49]
- $\kappa_6 = \tau_1(A^{\#})$, Seneta [50]
- $\kappa_7 = \frac{min_j ||A^{-1}_{(j)}||_\infty}{2}$, Ipsen and Meyer [39]

In particular, in the computation of some conditions numbers presented here above, the *matrix norm* is used. More precisely, it is possible to redefine the perturbation bound as follows:

$$\|\pi - \widetilde{\pi}\|_p \leq \kappa \|E\|_q$$

where $(p, q) = (\infty, \infty)$. Recall that the *matrix norm* on the vector space $K^{m\times n}$ is defined as follows:

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|.$$

Also Solan and Vieille, the authors of [52], have studied the effects of perturbations of the transition matrix on the stationary distribution. Their paper analysed the effects of perturbations of the transition matrix on the stationary distribution, by defining new method of measuring the difference between the two steady-state distribution vectors. In the following lines, part of results presented in [52] is reported.
Consider the set of states $\mathcal{S}$, containing at least two elements. C is a subset of $\mathcal{S}$ and $\bar{C} = \mathcal{S} \setminus C$ is the complement of C in $\mathcal{S}$. $\boldsymbol{Q}$ is the transition matrix over $\mathcal{S}$, and $\hat{\boldsymbol{Q}}$ is another transition matrix. $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$ are stationary distributions that correspond to $\boldsymbol{Q}$ and $\hat{\boldsymbol{Q}}$ respectively.
Hereafter, for every $C \subseteq S$ the next notation will be used: $\pi_C = \sum_{s\in C} \pi_s$.
And now, consider the next quantity:

$$\zeta_q = \min_{\emptyset \subset C \subset S} \sum_{s\in C} \pi_s \; q_{[\bar{C}]s}$$

which is the lowest among average frequency of transitions out of C. It can be considered as a measure of how isolated a subset C may be.

By using the quantity ζ_q defined just before, the concept of (ϵ, β)-*closeness* can be defined as follows:

Definition 2.3.1. *Let $\epsilon, \beta > 0$. A transition matrix $\hat{Q}$ is ϵ, β-close to Q if for every two states $i, j \in \mathcal{S}$,*

$$|1 - \frac{\hat{q}_{ij}}{q_{ij}}| \leq \beta$$

whenever

- $\pi_s q_{ij} \geq \epsilon\zeta_q$
- $\pi_s \hat{q}_{ij} \geq \epsilon\zeta_q$.

Additionally, the following quantity is defined, which will be incorporated in the next theorem:

$$L = \sum_{n=1}^{|\mathcal{S}|-1} \binom{|\mathcal{S}|}{n} n^{|\mathcal{S}|}$$

The next theorem summarizes all of the quantities introduced before and shows the measure of difference between the two different steady-state distribution.

Theorem 1. *Let* $\beta \in (0, 1/2^{|\mathcal{S}|})$ *and let*

$$\epsilon \in (0, \frac{\beta(1-\beta)}{L|\mathcal{S}|^4}).$$

For every irreducible transition matrix $\boldsymbol{Q}$ *on* $\mathcal{S}$ *and every transition matrix* $\hat{\boldsymbol{Q}}$ *that is* (ϵ, β)*-close to* $\boldsymbol{Q}$*,*

1. $\hat{\boldsymbol{Q}}$ is irreducible
2. its stationary distribution $\hat{\boldsymbol{\pi}}$ satisfies

$$|a - \frac{\hat{\pi_s}}{\pi_s}| \leq 18\beta L$$

for each $s \in \mathcal{S}$*.*

In this way, the authors of [52] provided bounds for the difference between the two steady-state distribution vectors coming from two different chains, namely the original and the perturbed one. Many authors have studied and analysed these bounds and each of them is useful in some context, in other situations other measures might be preferable. The choice of the most suitable bound is left to the user, who is going to analyse the Markov chain taken into consideration.

2.3.1 Quasi-Lumpability and Proportional Lumpability

In order to tackle the problem of state-state explosion, the approach of aggregating the state-space of the original Markov process can be considerably useful, as briefly mentioned in the previous sections. In particular, we already have seen that if the Markov chain is strongly lumpable, then the aggregated chain will behave exactly as the original model. However, *strong lumpability* consists in a quite strict condition, since it requires that for any two states within an equivalence class, their aggregated transition rates to any other class are the same. Indeed, in general a non-trivial lumpable partition might not exist.

An attempt to relax the conditions of strong lumpability in order to aggregate the states by also maintaining Markov properties properly, a new notion of lumpability called *quasi-lumpability* has been introduced in [45]. In the following lines, the definition present in the mentioned paper is reported.

Definition 2.3.2. (Quasi lumpability) *Let $X(t)$ be a CTMC characterized by the state space $\mathcal{S} = \{0, 1, ..., n\}$. Additionally, consider the equivalence relation $\sim$ over $\mathcal{S}$. We say that $X(t)$ is* quasi lumpable *with respect to $\sim$ (respectively, $\sim$ is a quasi lumpability for $X(t)$) if $\sim$ induces a partition on the state space of $X(t)$ such that for any equivalence class $[l], [k] \in \mathcal{S}/\sim$ with $[l] \neq [k]$ and $i, j \in [l]$,*

$$|q_{i[k]} - q_{j[k]}| \leq \epsilon, \epsilon \geq 0.$$

The notion of *quasi lumpability* consists in a generalization of the concept of lumpability, since it can be applied to a wider set of models of real world situations, by defining upper and lower bounds on the interested measures, like the steady-state probability vector. In particular, it allows to make the interested Markov chains lumpable by a relatively small perturbation to the transition rates. Moreover, the notion of quasi-lumpability coincides with the concept of *near-lumpability* discussed in [13]. Techniques for computing bounds to the steady state probabilities of quasi-lumpable Markov chains have been studied in [15, 28].

In this thesis another notion of lumpability, named *proportional lumpability* will be widely used. Proportional lumpability extends the original definition of strong lumpability. It consists in a relaxed version of strong lumpability but differently from quasi-lumpability, it allows one to derive an exact solution to the original process, instead of computing only upper and lower bounds. In the following lines, the definition of proportional lumpability is reported:

Definition 2.3.3. (Proportional Lumpability) *Let $X(t)$ be a CTMC characterized by the state space $\mathcal{S} = \{0, 1, ..., n\}$. Additionally, consider the equivalence relation $\sim$ over $\mathcal{S}$. We say that $X(t)$ is* proportionally lumpable *with respect to $\sim$ (respectively, $\sim$ is a proportional lumpability for $X(t)$) if there exists a function κ from $\mathcal{S}$ to $\mathbb{R}^+$ such that $\sim$ induces a partition on the state space of $X(t)$ satisfying the property that for any equivalence class $[l], [k] \in \mathcal{S}/\sim$ with $l \neq k$ and $i, j \in [l]$,*

$$\frac{q_{i[k]}}{\kappa(i)} = \frac{q_{j[k]}}{\kappa(j)}.$$

$X(t)$ *is* κ-proportionally lumpable *with respect to* $\sim$ *(respectively,* $\sim$ *is a* κ*-proportional lumpability for* $X(t)$*) if* $X(t)$ *is* proportionally lumpable *with respect to* $\sim$ *and function* κ*.*

The key result on the proportional lumpability consists in the possibility of computing the exact solution of the original model from the aggregated process. Consider the next proposition:

Proposition 3. (*Aggregated process for proportional lumpability*) *Let us consider the CTMC* $X(t)$ *characterized by the state space* $\mathcal{S}$*, infinitesimal generator matrix* $\boldsymbol{Q}$ *and equilibrium distribution* $\boldsymbol{\pi}$*. Let* κ *be a function from* $\mathcal{S}$ *to* $\mathbb{R}^+$*,* $\sim$ *be a* κ*-proportional lumpability for* $X(t)$ *and* $\widetilde{X}(t)$ *be the aggregated process with state space* $\mathcal{S}/\sim$ *and infinitesimal generator* $\widetilde{\boldsymbol{Q}}$ *defined by: for any equivalence class* $[l], [k] \in \mathcal{S}/\sim$ *with* $l \neq k$

$$\widetilde{q}_{[l][k]} = \frac{q_{i[k]}}{\kappa(i)}$$

for any $i \in [l]$*. Then the invariant measure* $\widetilde{\boldsymbol{\mu}}$ *of* $\widetilde{X}(t)$ *satisfies: for any equivalence class* $[s] \in \mathcal{S}/\sim$*,*

$$\widetilde{\mu}([s]) = \sum_{s \in [s]} \pi(s)\kappa(s).$$

Proof. We denote with $\widetilde{X}(t)$ the aggregated process defined in previous lines. For all $[s] \in \mathcal{S}/\sim$, the corresponding global balance equation can be expressed as follows:

$$\widetilde{\mu}([s]) \sum_{\substack{[s'] \in \mathcal{S}/\sim \\ s' \neq s}} \widetilde{q}_{[s][s']} = \sum_{\substack{[s'] \in \mathcal{S}/\sim \\ s' \neq s}} \widetilde{\mu}([s'])\widetilde{q}_{[s'][s]}. \tag{2.2}$$

We can proceed with the proof by substituting $\widetilde{q}$ and $\widetilde{\mu}$ according to the definitions given above. As we can see easily, the left-hand side of Equation 2.2 evolves as follows, where s is an arbitrary state in $[s]$:

$$\left(\sum_{s \in [s]} \pi(s)\kappa(s) \right) \sum_{\substack{[s'] \in \mathcal{S}/\sim \\ s' \neq s}} \frac{q_{s[s']}}{\kappa(s)} =$$

$$\sum_{s \in [s]} \pi(s) \sum_{\substack{[s'] \in \mathcal{S}/\sim \\ s' \neq s}} \sum_{s' \in [s']} q_{ss'}.$$

Equivalently, the right-hand side of 2.2 will be:

$$\sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \left(\sum_{s'\in[s']} \pi(s')\kappa(s') \right) \frac{q_{s'[s]}}{\kappa(s')} = \sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \sum_{s'\in[s']} \pi(s') \sum_{s\in[s]} q_{s's} = \sum_{s\in[s]} \sum_{\substack{[s']\in\mathcal{S}/\sim \\ s'\neq s}} \sum_{s'\in[s']} \pi(s') q_{s's}.$$

According to the general conservation law, the effective flow inward must equal the effective flow outward, for any closed boundary, meaning that for any $S \subseteq \mathcal{S}$

$$\sum_{s\in S} \pi(s) \sum_{s'\in\mathcal{S}, s'\notin S} q_{ss'} = \sum_{s\in S} \sum_{s'\in\mathcal{S}, s'\notin S} \pi(s') q_{s's}$$

and this concludes the proof. □

Therefore, the steady-state equilibrium distribution of a proportionally lumpable CTMC $X(t)$ can be computed from the steady-state equilibrium distribution of a class of perturbations $X'(t)$. Consider the next definition:

Definition 2.3.4. (*Perturbation w.r.t. κ and $\sim$*) *Let us consider the CTMC $X(t)$ characterized by the state space $\mathcal{S}$, and infinitesimal generator matrix $\boldsymbol{Q}$. Let κ be a function from $\mathcal{S}$ to $\mathbb{R}^+$ and $\sim$ be a κ-proportional lumpability for $X(t)$. We say that a CTMC $X'(t)$ with infinitesimal generator matrix $\boldsymbol{Q'}$ is a perturbation of $X(t)$ with respect to κ and $\sim$ if $X'(t)$ is obtained from $X(t)$ by perturbing its rates such that for all $s \in \mathcal{S}$, $[s] \in \mathcal{S}/\sim$,*

$$\sum_{\substack{s'\in[s]/\sim \\ s'\neq s}} q'_{ss'} = \frac{\sum_{\substack{s'\in[s]/\sim \\ s'\neq s}} q_{ss'}}{\kappa(s)}.$$

Proposition 4. (*Equilibrium distribution for proportionally lumpable CTMCs*) *Let us consider the CTMC $X(t)$ characterized by the state space $\mathcal{S}$, and infinitesimal generator matrix $\boldsymbol{Q}$ and equilibrium distribution $\boldsymbol{\pi}$. Let κ be a*

function from $\mathcal{S}$ to $\mathbb{R}^+$, $\sim$ be a κ-proportional lumpability for $X(t)$ and $\widetilde{X}(t)$ be the aggregated process with state space $\mathcal{S}/\sim$ and infinitesimal generator $\widetilde{\boldsymbol{Q}}$ as defined in Proposition 3. Then, for any perturbation $X'(t)$ of the original chain $X(t)$ with respect to κ and $\sim$ according to Definition 2.3.4, the equilibrium distribution $\boldsymbol{\pi'}$ of $X'(t)$ satisfies the following property: let $K = \sum_{s\in\mathcal{S}} \pi'(s)/\kappa(s)$ then

$$\pi(s) = \frac{\pi'(s)}{K\kappa(s)}.$$

Proof. For all $s \in \mathcal{S}$, the corresponding global balance equation is

$$\pi(s) \sum_{\substack{s'\in\mathcal{S}\\ s'\neq s}} q_{ss'} = \sum_{\substack{s'\in\mathcal{S}\\ s'\neq s}} \pi(s') q_{s's}. \tag{2.3}$$

Since $\sim$ induces a partition on the state space of $X(t)$, the equation can be expressed as:

$$\begin{aligned} &\pi(s)\Bigg(\sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{s'\in[s]} q_{ss'} + \sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{\substack{s'\in[s]\\ s'\neq s}} q_{ss'} \Bigg) = \\ &\sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{s'\in[s]} \pi(s') q_{s's} + \sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{\substack{s'\in[s]\\ s'\neq s}} \pi(s') q_{s's}. \end{aligned} \tag{2.4}$$

The definition of $\pi(s)$ reported in previous lines can be exploited so that the left-hand side of Equation 2.4 can be written as follows:

$$\begin{aligned} &\frac{\pi'(s)}{K\kappa(s)}\Bigg(\sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{s'\in[s]} q_{ss'} + \sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{\substack{s'\in[s]\\ s'\neq s}} q_{ss'} \Bigg) = \\ &\frac{\pi'(s)}{K}\Bigg(\sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{s'\in[s]} \frac{q_{ss'}}{\kappa(s)} + \sum_{\substack{[s]\in\mathcal{S}/\sim\\ s\neq[s]}} \sum_{\substack{s'\in[s]\\ s'\neq s}} \frac{q_{ss'}}{\kappa(s)} \Bigg) = \\ &\frac{\pi'(s)}{K} \sum_{\substack{s'\in\mathcal{S}\\ s'\neq s}} q'_{ss'}. \end{aligned}$$

Equivalently, the right-hand side of Equation 2.4 evolves as follows:

$$\sum_{\substack{[s]\in\mathcal{S}/\sim \\ s\neq[s]}} \sum_{s'\in[s]} \frac{\pi'(s')}{K\kappa(s')} q_{s's} + \sum_{\substack{[s]\in\mathcal{S}/\sim \\ s\neq[s]}} \sum_{\substack{s'\in[s] \\ s'\neq s}} \frac{\pi'(s')}{K\kappa(s')} q_{s's} =$$

$$\frac{1}{K} \sum_{\substack{[s]\in\mathcal{S}/\sim \\ s\neq[s]}} \sum_{s'\in[s]} \pi'(s') \frac{q_{s's}}{\kappa(s')} + \frac{1}{K} \sum_{\substack{[s]\in\mathcal{S}/\sim \\ s\neq[s]}} \sum_{\substack{s'\in[s] \\ s'\neq s}} \pi'(s') \frac{q_{s's}}{\kappa(s')} =$$

$$\frac{1}{K} \sum_{\substack{s'\in\mathcal{S} \\ s'\neq s}} \pi'(s') q'_{s's}.$$

Therefore, for all $s \in \mathcal{S}$ the global balance equation of $X'(t)$ is satisfied, meaning that we can obtain:

$$\pi'(s) \sum_{\substack{s'\in\mathcal{S} \\ s'\neq s}} q'_{ss'} = \sum_{\substack{s'\in\mathcal{S} \\ s'\neq s}} \pi'(s') q'_{s's}.$$

The normalizing condition is satisfied too: i.e., $\sum_{s\in\mathcal{S}} \pi(s) = 1$. The proof follows trivially from the fact that $K = \sum_{s\in\mathcal{S}} \pi'(s)/\kappa(s)$, in fact:

$$\sum_{s\in\mathcal{S}} \pi(s) = \sum_{s\in\mathcal{S}} \frac{\pi'(s)}{K\kappa(s)} = \frac{1}{K} \sum_{s\in\mathcal{S}} \frac{\pi'(s)}{\kappa(s)} = \frac{1}{K} K = 1.$$

□

Example 1. Consider the CTMC $X(t)$ of the Figure 2.4 with $\rho \neq \nu$. Let $\mathcal{S} = \{1, 2, 3, 4\}$ be its state space and $\sim$ be the equivalence relation such that $1 \sim 3$ and $2 \sim 4$ inducing the partition $\mathcal{S}/\sim = \{[1, 3], [2, 3]\}$. $\sim$ is a strong lumpability for the considered Markov chain because $q_{1[2,4]} = \lambda + \mu$ which is equal to $q_{3[2,4]} = \lambda + \mu$ and similarly, $q_{2[1,3]} = \rho + \nu$ is equal to $q_{4[1,3]} = \rho + \nu$.

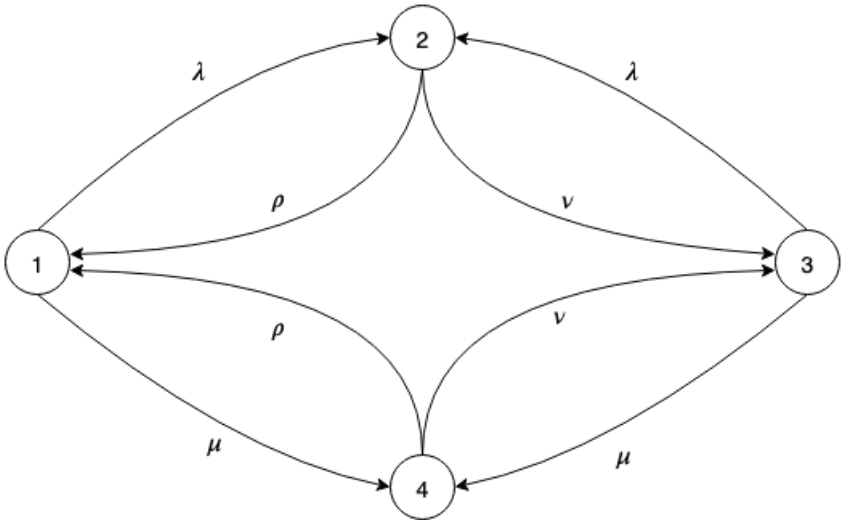

Figure 2.4: Strongly lumpable CTMC

Let κ be a function from $\mathcal{S}$ to $\mathbb{R}^+$ such that $\kappa_1 = 2$, $\kappa_2 = 1/2$, $\kappa_3 = 3$ and $\kappa_4 = 4$. Now, in order to make the chain proportionally lumpable, we introduce different rates for each transition as depicted in Figure 2.5: we set $\lambda = 0.15$, $\mu = 0.35$, $\eta = 0.7$, $\gamma = 0.05$, $\rho = 0.05$, $\nu = 0.025$, $\epsilon = 0.2$ and $\delta = 0.4$.
The chain $X(t)$ can be said κ-proportionally lumpable because

$$\frac{q_{1[2,4]}}{\kappa_1} = \frac{q_{3[2,4]}}{\kappa_3}$$

since $\frac{q_{1[2,4]}}{\kappa_1} = \frac{\lambda+\mu}{2}$ and $\frac{q_{3[2,4]}}{\kappa_3} = \frac{\eta+\gamma}{3}$. If we substitute the rates we obtain $\frac{0.15+0.35}{2} = \frac{0.7+0.05}{3} = 0.25$.

Similarly, we can prove that

$$\frac{q_{2[1,3]}}{\kappa_2} = \frac{q_{4[1,3]}}{\kappa_4}$$

since $\frac{q_{2[1,3]}}{\kappa_2} = \frac{\rho+\nu}{1/2}$ and $\frac{q_{4[1,3]}}{\kappa_4} = \frac{\epsilon+\delta}{4}$. If we substitute the rates we obtain $2(0.05 + 0.025) = \frac{0.2+0.4}{4} = 0.15$.

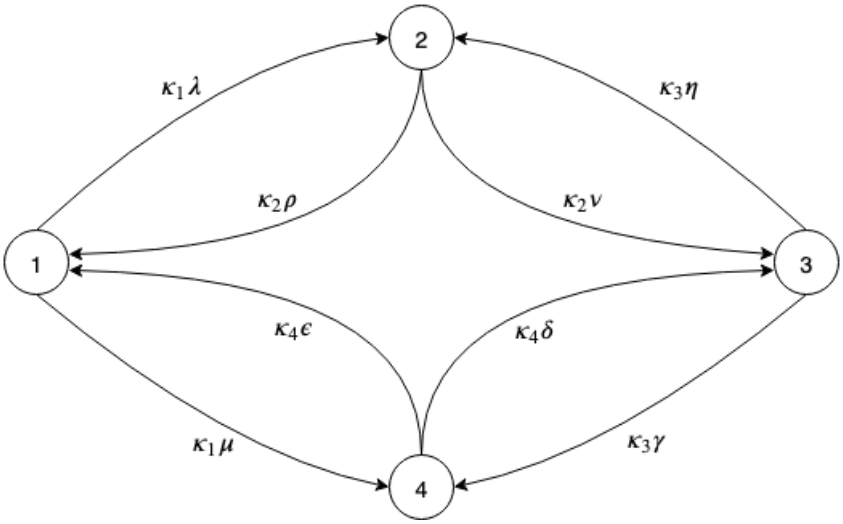

Figure 2.5: Proportionally lumpable CTMC

Example 2. In this example, a system with multiple CPUs is considered. Each CPU is equipped with its own private memory and a common memory available only by one CPU at a time. The processors execute in private memory for some random duration and then they may issue a common memory access request. Assume that this random time is exponentially distributed with parameter λ_P for processor P. The common memory access duration is also assumed to be exponentially distributed with parameter μ_P for processor P (i.e., the average duration of a common memory access is $1/\mu_P$).
In this example, a system with two processors A and B is considered. Assume that the processors are characterized by different distribution parameters: the private and common memory accesses of A are governed by two exponential distributions with parameters λ_A and μ_A, respectively, while the private and common memory accesses of B are governed by two exponential distributions with parameters λ_B and μ_B, respectively. The CTMC describing the behaviour of this two-processor system is depicted in Figure 2.6, it has five states as follows:

- State 1:
  - processor A and processor B both executing in their private memories;
- State 2:
  - processor B executing in private memory;
  - processor A accessing common memory;
- State 3:
  - processor A executing in private memory;
  - B accessing common memory;
- State 4:
  - processor A accessing common memory;
  - processor B waiting for common memory;
- State 5:
  - processor B accessing common memory;
  - processor A waiting for common memory.

Suppose that the rates are related as follows:

$$\lambda_A = k_1\lambda \quad \lambda_B = k_2\lambda \quad \mu_A = k_2\mu \quad \mu_B = k_1\mu$$

for $\lambda, \mu, k_1, k_2 \in \mathbb{R}^+$. In this case the CTMC appears as represented in Figure 2.7. It is possible to observe that is proportionally lumpable with respect to the equivalence classes $S_1 = \{1\}, S_{2,3} = \{2, 3\}$ and $S_{4,5} = \{4, 5\}$ and the function κ defined by: $\kappa(1) = 1$, $\kappa(2) = k_2$, $\kappa(3) = k_1$, $\kappa(4) = k_2$ and $\kappa(5) = k_1$. We can then analyse the reduced chain represented in Figure 2.8 and the exact solution of the original model can be computed by using Propositions 3 and 4.

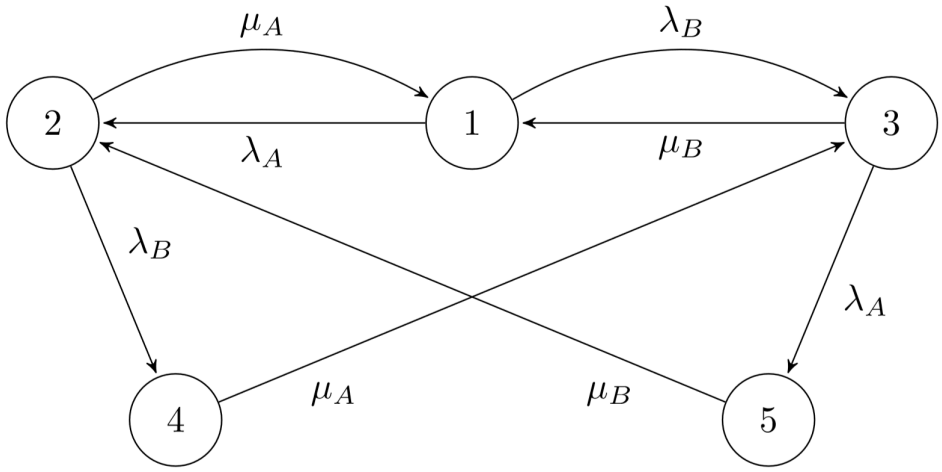

Figure 2.6: Two processor system

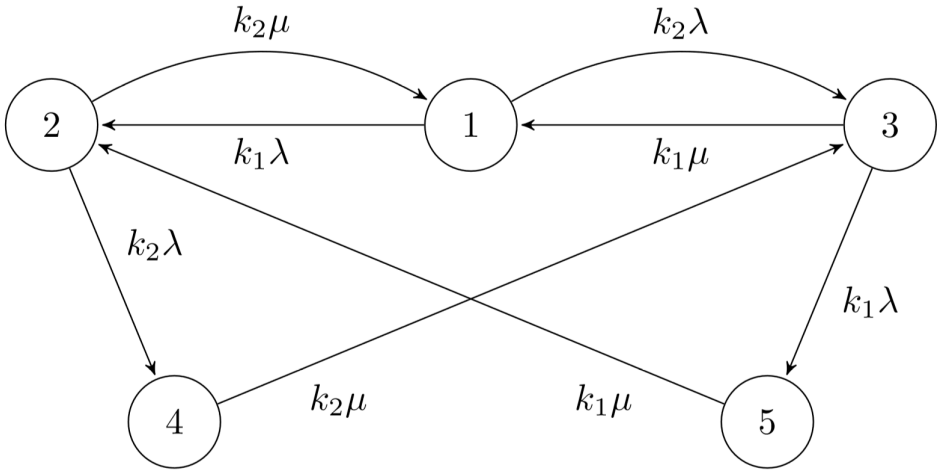

Figure 2.7: Two processor system with proportional factors

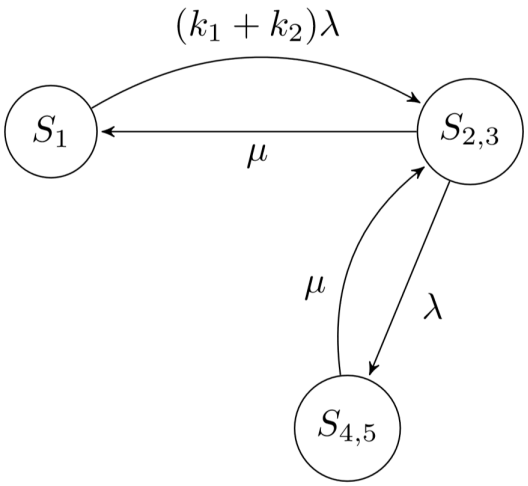

Figure 2.8: Two processor reduced system

Chapter 3

Introduction to PEPA

3.1 Overview

PEPA, acronym of *Performance Evaluation Process Algebra*, is a stochastic process algebra designed for modelling computer and communication systems composed of concurrently components which cooperate and share work. The PEPA project has started in Edinburgh in 1991, when Jane Hilston defined the PEPA language in her PhD thesis, undertaken in the Laboratory for Foundations of Computer Science, a research institute of the Division of Informatics.

A simple computer system can be easily modelled without any definition of some suitable language. However, as computer and communication systems become more large and complex, also their models become more complicated and therefore, in this context, the need to express them through some high-level expressive formal language has arisen, in order to perform performance analysis on them. Jane Hillston's *Performance Evaluation Process Algebra (PEPA)* is then a formal language for modelling such systems in order to understand the dynamic behaviour of a system with respect to dynamic properties such as throughput and response time. PEPA models are built by building-brick components which exhibit individual activities or alternatively they can also cooperate on shared activities with other components, and an estimate of the rate at which an activity may be performed is associated to each activity. Other process algebras mentioned in the introduction such as *CCS* and *CSP* offer a compositional description technique as PEPA, but they do not include timing information for performance estimation. *Performance Evaluation Process Algebra* has been elaborated for the purpose of introducing a new process algebra

suitable for performance analysis.

Since PEPA models contain information about the rate associated to each activity, which is drawn from the exponential distribution, they allow one to generate a corresponding continuous time Markov chain (CTMC) and the model can be solved in terms of steady-state equilibrium behaviour by using linear algebra. It can be summarized that *Performance Evaluation Process Algebra*(PEPA) is an algebraic calculus enhanced with stochastic timing information which may be used to calculate performance measures characterizing the system, necessary for performance evaluation.

3.2 Syntax of PEPA

In this section the fundamental syntax of the PEPA language is presented. Notice that the notations which have been used in this section and through all this work are those used by Jane Hillston herself in her book [35].

The basic elements (primitives) of the language PEPA are *components* and *activities*. Components are active elements within a system, activity captures actions of those units. With the term *cooperation* the interaction between two different components is indicated. PEPA models are constructed from components which engage in activities. For each activity in PEPA there exists an associated duration. The duration is a random variable governed by an exponential distribution which is characterized by a unique parameter, the duration of an activity, which is represented by a single real number. The duration parameter of each activity is called *activity rate*: it may be any positive real number or the distinguished symbol $\top$. $\top$ indicates the rate as *unspecified*.
Each activity is represented as a pair (α, r) where α is the action type of the activity, r is the activity rate. In each system an action is assumed to be uniquely typed and there is assumed to be a countable set $\mathcal{A}$ of all possible action types: $\mathcal{A}$ is then called the set of action types. Activities with the same action type represent several occurrences of that action performed by the system. There exists a special action type, denoted τ and named *unknown type*: it represents an unknown or unimportant system action. Activities characterized by the unknown action type will be concealed to the component in which they occur.

To sum up the explanation given here above:

- $\mathcal{A}$: set of all action types, including τ
- R^+: set of all positive real numbers, including $\top$
- $Act = \mathcal{A} \times R^+$: set of all activities

Hence, an activity is represented as:

$$a = (\alpha, r)$$

where

- $a \in Act = \mathcal{A} \times R^+$ denotes the activity;
- $\alpha \in \mathcal{A}$ is the action type;
- $r \in R^+$ is the activity rate.

The exhibition of the activity $a = (\alpha, r)$ implies a delay, whose duration is governed by its characterizing probability distribution function. The probability that the activity $a = (\alpha, r)$ happens within a period of time of length t is:

$$F_a(t) = 1 - e^{-rt}$$

In literature, usually components are denoted by capital letters, such as $P, C, C_i, ...$; activities are denoted by lowercase letters such as $a, b, c, ...$; action types are denoted by Greek letters such as $\alpha, \beta, \gamma, ...$ or names like *task*, *request*, *use*,...; and finally, activity rates are denoted by $r, s, t, r_i, ...$ or sometimes also Greek letters such as μ and λ are used.

3.2.1 PEPA operators

In this section the PEPA language specification will be presented, by explaining the characteristics of each combinator.
The following grammar defines the syntax for PEPA terms:

$$P ::= P \underset{L}{\bowtie} P \mid P/L \mid S$$
$$S ::= (\alpha, r).S \mid S + S \mid A$$

where S denotes a *sequential component*, while P denotes a model component which executes in parallel. We assume that there is a countable set of *constants*, A. The set of all possible components within the considered system is denoted by $\mathcal{C}$.

Prefix
$(\alpha, r).S$ means that the component subsequently behaves as component S. It consists in the basilar mechanism by which the behaviour of components are determined.
The component $(\alpha, r).S$ is said to be *carrying out* the activity (α, r) with some action type α and a duration which is exponentially distributed with parameter r. Therefore, the duration of the activity, namely the time taken to complete the activity, will be some Δt, drawn from the distribution which governs the duration random variable. If the system is in the state $(\alpha, r).S$ at some time t, the time at which it completes (α, r) and becomes S will be $t + \Delta t$. In case $a = (\alpha, r)$ the component $(\alpha, r).P$ can be written as $a.P$.

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

Choice
As the name suggests, the component $P + Q$ represents a system which can possibly exhibit the component P's actions or alternatively the component Q's actions. Therefore, the component $P + Q$ is going to empower all the current activities of P and all the current activities of Q. This kind of situation is obtained when P and Q are competing for the same resource they want both to use, so that the choice operation is representing a competition between components. The decision on which is going to perform is made as follows: the first activity to complete distinguishes one of the components P or Q. The other component of the choice is discarded. Whichever enabled activity completes, it must clearly belong to either P or Q.
Suppose that $P = (\alpha, r).P'$ and $Q = (\beta, s).Q'$. At the time t_0 the enabled activities of $P + Q$ are both α, r and (β, s). Let Δ_α and Δ_β drawn from the exponential distribution of α and β, respectively. For $x \in \{\alpha, \beta\}$:

- Δ_x represents the time taken for the activity x;
- $F_x(t)$ is the probability that $\Delta_x \leq t$.

If $\Delta_\alpha < \Delta_\beta$ then activity (α, r) is enabled and at the time $t_0 + \Delta_\alpha$ the system behaves as P'.
Notice that the probability that $\Delta_\alpha = \Delta_\beta$ is 0. Indeed, notice that we are dealing with continuous probability distributions and it is well known that the

probability that P and Q both completing an activity at the same time is 0.

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$

Cooperation
The cooperation combinator is applied as follows: $P \underset{L}{\bowtie} Q$.
Cooperation is not a unique combinator, but rather it can be recognized as an indexed set of them, one combinator for each possible set of action types $L \subseteq \mathcal{A}$. L is named *cooperation set* and defines the action types on which the components must synchronize or better *cooperate*, that is the interaction between the components. While the choice combinator is representing a competition between components, the cooperation combinator is going to synchronize and proceed together with the other component. Indeed, in a cooperation context each component is assumed to have its own implicit resource and they proceed independently with any activities whose types do not occur in the cooperation set. Therefore, all activities of P and Q which have types which do not occur in L will proceed unaffected. These are called *individual* activities. Activities with action types in L are called *shared* activities and they require the involvement of both components in conjunction in an activity of that type. In this situation, a component can be forced to be blocked in order to wait for the other component to be ready to participate. These activities represent situations in the system when the components need to work together to achieve an action.
If an activity has an unspecified rate in a component, then the component is said to be *passive* with respect to that action type, and it does not contribute to the work involved. The unknown action type τ may not appear in any cooperation set, i.e., $\tau \notin L$.

To summing up, there exist two types of activities, which are:

- *individual activities*: activities of P and Q whose action types do not occur in L;

- *shared activities*: activities of P and Q whose action types do occur in L. They will only be enabled in $P \underset{L}{\bowtie} Q$ when they are enabled in both P and Q. Shared activities need to work together to achieve an action. Thus one component may become blocked, waiting for the other component to be ready to participate.

When two shared activities cooperate, a new shared activity should be considered, which is formed by the cooperation. This activity will have the same action type as the two contributing activities and a rate reflecting the rate of the slower component. The expected duration of a shared activity will be greater than or equal to the expected durations of the corresponding activities in the cooperating components.
Notice that, the special case in which $L = \emptyset$, namely the cooperation set is an empty set, then the cooperation operator has the effect of *parallel composition*, allowing components to proceed concurrently without any interaction between them. We use the notation $P \parallel Q$ to represent $P \underset{\emptyset}{\bowtie} Q$ where $\parallel$ is the parallel combinator.

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P' \underset{L}{\bowtie} Q} \ (\alpha \notin L)$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P \underset{L}{\bowtie} Q'} \ (\alpha \notin L)$$

$$\frac{P \xrightarrow{(\alpha,r_1)} P' \ Q \xrightarrow{(\alpha,r_2)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,R)} P' \underset{L}{\bowtie} Q'} \qquad R = \frac{r_1}{r_\alpha(P)} \frac{r_2}{r_\alpha(Q)} \ min(r_\alpha(P), r_\alpha(Q)) \ (\alpha \in L)$$

Hiding
P/L is as like as P, saving the activities of type belonging to the set L which are concealed. This means that these concealed, or better, hidden types appear as the unknown type τ, and they can be considered as an internal delay by the component. Normally, when an activity is completed an external observer can see the type of the completed activity and additionally, the observer is also aware of the delay while the activity takes place. A concealed activity is

considered only as a time delay and the action type is the unknown type τ. Notice that:

- The action type of a hidden activity cannot be seen by an observer or another component;
- The duration of an activity remains unaffected.

Moreover such an activity cannot be carried out in cooperation with any other component.

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\alpha,r)} P'/L} \ (\alpha \notin L)$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\alpha,r)} P'/L} \ (\alpha \in L)$$

Constant

Initially we have assumed that there is a countable set of constants A. Constants are components whose defintion is given by a writing down an equation such as $A \stackrel{\text{def}}{=} P$. If $A \stackrel{\text{def}}{=} P$ then A denotes a component behaving as P. This is how we assign names to components.

$$\frac{P \xrightarrow{(\alpha,r)} P'}{A \xrightarrow{(\alpha,r)} P'} \ (A \stackrel{\text{def}}{=} P)$$

The semantics of the language, presented in structured operational semantics style, are shown in the table here below. The transitional semantics over PEPA is then given by the least multi-relation $\rightarrow \subseteq \mathit{PEPA} \times \mathcal{A}ct \times \mathit{PEPA}$ satisfying the rules.

$$\frac{}{(\alpha,r).P \xrightarrow{(\alpha,r)} P} \qquad \frac{P \xrightarrow{(\alpha,r)} P'}{P+Q \xrightarrow{(\alpha,r)} P'} \qquad \frac{Q \xrightarrow{(\alpha,r)} Q'}{P+Q \xrightarrow{(\alpha,r)} Q'}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\alpha,r)} P'/L} \quad (\alpha \notin L) \qquad \frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\alpha,r)} P'/L} \quad (\alpha \in L)$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{A \xrightarrow{(\alpha,r)} P'} \quad (A \stackrel{\text{def}}{=} P)$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P' \underset{L}{\bowtie} Q} \quad (\alpha \notin L) \qquad \frac{Q \xrightarrow{(\alpha,r)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P \underset{L}{\bowtie} Q'} \quad (\alpha \notin L)$$

$$\frac{P \xrightarrow{(\alpha,r_1)} P' \quad Q \xrightarrow{(\alpha,r_2)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,R)} P' \underset{L}{\bowtie} Q'} \quad R = \frac{r_1}{r_\alpha(P)} \frac{r_2}{r_\alpha(Q)} \quad min(r_\alpha(P), r_\alpha(Q)) \ (\alpha \in L)$$

Table 1: Operational semantics for PEPA components

The precedence of PEPA operators is defined as follows:

1. Hiding
2. Prefix
3. Cooperation
4. Choice

When brackets are missing, we assume that the cooperation associates to the left, i.e., $P \underset{L_1}{\bowtie} Q \underset{L_2}{\bowtie} R$ behaves as $(P \underset{L_1}{\bowtie} Q) \underset{L_2}{\bowtie} R$. The cooperation between several different components may be regarded as being built up in layers or levels, each cooperation combining just two components.

The semantics of each term in PEPA is given via a labelled *multi-transition system* where the multiplicities of arcs are significant. A *labelled transition system* is a triple $(S, T, \{\xrightarrow{\mathrm{t}} \mid t \in T\})$ where

- S is a set of states;
- T is a set of transition labels;
- $\xrightarrow{\mathrm{t}} \subseteq S \times S$ is a transition relation for each $t \in T$.

In the transition system, a state corresponds to the evolution of one component into another. The set of reachable states of a model P is called *derivative set* of P, denoted as $ds(P)$, and is defined as follows:

- if $P \stackrel{\mathrm{def}}{=} P_0$ then $P_0 \in ds(P)$;
- if $P_i \in ds(P)$ and there exists $a \in Act(P_i)$ such that $P_i \xrightarrow{\mathrm{a}} P_j$ then $P_j \in ds(P)$.

$ds(P)$ constitutes the set of nodes of the derivation graph of P ($\mathcal{D}(P)$) obtained by applying the semantic rules exhaustively. For any component P the *exit rate* from P will be the sum of the activity rates of all activities enabled in P, i.e., $q(P) = \sum_{a \in \mathcal{A}ct} r_a$ with r_a being the rate of activity a. If P enables more than one activity, so $|\mathcal{A}ct| > 1$ then the dynamic behaviour of the model is determined by a race condition.

Race condition
A *race condition* determines the dynamic behaviour of a model when more than one activity is involved: this is the situation in which many activities attempt to proceed but only the fastest succeeds, the choice combinator case. Naturally, which activity is fastest will vary due to the nature of the random variables determining the duration of activities. The probability that a particular activity completes will be given by the ratio of the activity rate of that activity to the sum of the activity rates of all the enabled activities.

Building the derivation graph is the basis of the construction of the underlying Continuous Time Markov Chain (CTMC). A state of the chain is associated with each component of the derivative set $ds(P)$ and the transitions between states are derived from the arcs of the derivation graph. Consider the following theorem:

Theorem 2. *For any finite PEPA model* $P \stackrel{def}{=} P_0$ *with* $ds(P) = \{P_0, ..., P_n\}$, *if we define the stochastic process* $X(t)$, *such that* $X(t) = P_i$ *indicates that the system behaves as component* P_i *at time* t, *then* $X(t)$ *is a CTMC.*

The omitted proof can be found in Chapter 3 of [35].
The transition rate between two components P and Q is denoted by $q(P, Q)$ and it is the rate at which the system changes from behaving as component P to behaving as Q. Numerically it consists in the sum of the activity rates labelling arcs which connect P and Q in $\mathcal{D}(P)$.

Finally, let us consider an example of PEPA model, specifically from the PEPA model equations to the corresponding graphical representation.

Example 1. Consider a system composed by two components *Process* and *Resource*. This system is going to represent a simple resource usage as two cooperating components. The *Process* component will exhibit two activities consecutively: *use* and *task*; the *Resource* component will exhibit two activities consecutively too: *use* and *update*.

The PEPA model of the *Process* component is as follows:

$$Process \stackrel{\text{def}}{=} (use, r_1).Process'$$
$$Process' \stackrel{\text{def}}{=} (task, r_2).Process$$

The PEPA model of the *Resource* component is as follows:

$$Resource \stackrel{\text{def}}{=} (use, r_3).Resource'$$
$$Resource' \stackrel{\text{def}}{=} (update, r_4).Resource$$

And finally, the PEPA model of the entire system is

$$System \stackrel{\text{def}}{=} Process \underset{use}{\bowtie} Resource$$

Therefore, the transition diagram is depicted in the following Figure 3.1. Notice that $r_{13} = min(r_1, r_2)$.

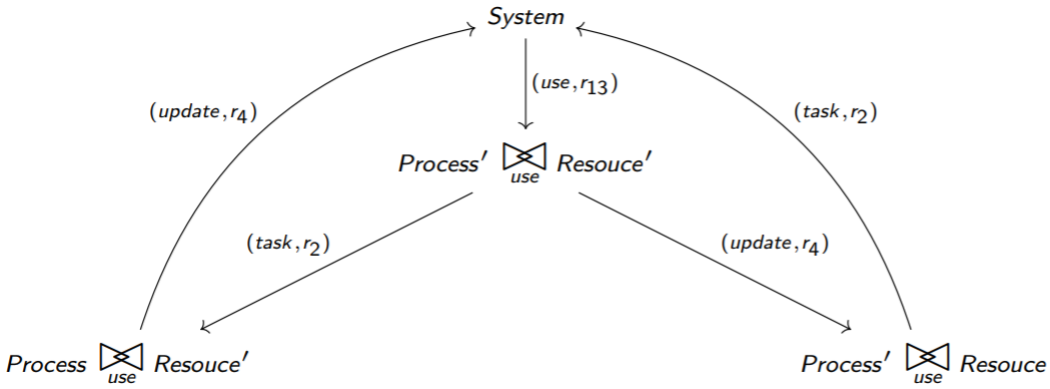

Figure 3.1: Transition diagram of the PEPA model considered in *Example 1*

Chapter 4

Notions of Equivalence

For the purpose of tackling the problem of large performance models, the techniques of model reduction and state-space aggregation of the underlying Markov process are presented in terms of equivalence relation. *Equivalence* is a criterion used to determine if two models can be considered to be indistinguishable. Notice that equivalence relations can be applied not only state-to-state, the case which we are interested in, but also system-to-model, used to establish the confidence in the model as a representation of the system being investigated, and model-to-model, used to compare models in order to find alternative representations of the system.
In this chapter, the concept of *bisimulation* for PEPA models will be studied and afterwards the notion of *strong equivalence* and *lumpable bisimulation* will be presented.

4.1 Bisimulation

The notion of *bisimulation* is based on the appearance of the externally observed behaviour. Specifically, two agents are considered to be bisimilar when their externally observed behaviour appears to be the same: indeed, if two agents are bisimilar, it is not possible to distinguish them by an external observer.
The notion of *strong bisimilarity* is defined in [35] as follows:

Definition 4.1.1. (Strong Bisimulatiton) *Let us consider the binary relation* $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ *, over PEPA components.* $\mathcal{R}$ *results to be a* strong bisimulation *if* $(P, Q) \in \mathcal{R}$ *implies, for all* $\alpha \in \mathcal{A}$, $r_\alpha(P) = r_\alpha(Q)$ *and for all* $a \in \mathcal{A}ct$,

1. *If* $P \xrightarrow{a} P'$ *then for some* Q', $Q \xrightarrow{a} Q'$, *and* $(P', Q') \in \mathcal{R}$;
2. *If* $Q \xrightarrow{a} Q'$ *then for some* P', $P \xrightarrow{a} P'$, *and* $(P', Q') \in \mathcal{R}$.

If P and Q are *strongly bisimilar* any action performed by one must be matched by the other, and any subsequent action must also be matched.

Formally, two components P and Q are strongly bisimilar if:

- Every a activity of a component correspond to a activity of the other;
- Every a-derivative of a component is strongly bisimilar to $a-$derivative of the corresponding component;
- For P and Q, for every action type the apparent rates are equal.

Notice that the intuitive idea of considering two systems equivalent if they have isomorphic labelled transition systems does not work, because there can exist that two systems with non-isomorphic labelled transition systems have the exactly same behaviour with each other. Consider the next example:

Example 1. Consider the following Figure 4.1 representing two different labelled transition systems. Even if they behave in the same way, the isomorphic equivalence criterion will distinguish them, and this is not what we want to obtain.

Figure 4.1: Two systems with different LTSs

Indeed, it can be easily seen that their behaviour is the same: they are going to execute only action named a infinitely, and therefore they must be considered equivalent.
Furthermore, we must specify that bisimulation is also different from other

equivalences over labelled transition systems introduced in literature like *traces equivalence* and *decorated-traces equivalence*. Simply put, in [19] is explained that traces equivalence put together two systems executing the identical sequences of actions, while decorated-traces equivalence considers equivalent two systems executing the identical sequences of actions and additionally it requires that two comparing systems are able to accept the same set of actions after each sequence. Consider the next example, in which the differences between these three notions of equivalence are shown graphically:

Example 2. Consider the following Figure 4.2, taken from the materials of the course *Formal Methods for System Verification* of *Ca' Foscari University of Venice* by Prof. Sabina Rossi, representing three different coffee/tea machines. They accept only two types of coins: *coin 1* and *coin 2*. The traces equivalence will consider equivalent all of them, while the decorated-traces equivalence will put together the second and the third one, rather the bisimulation based equivalence will consider all of them as different.

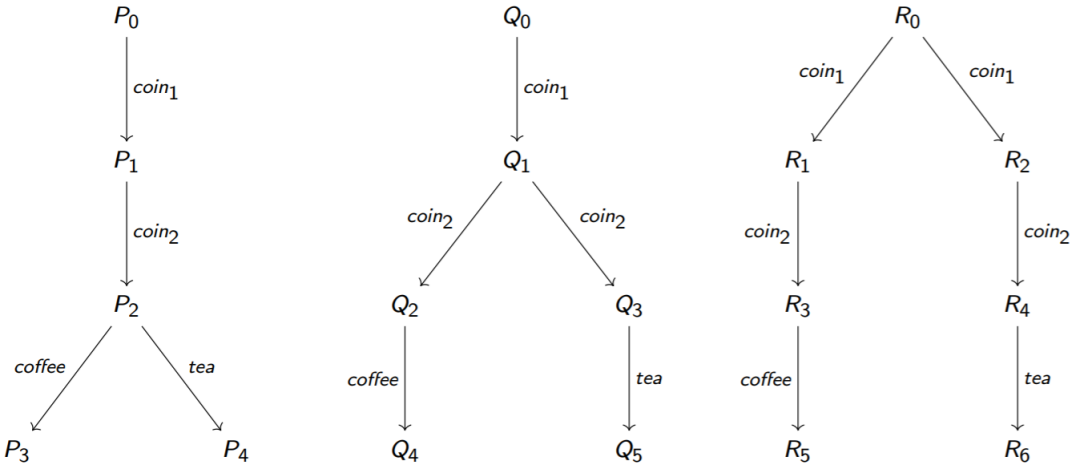

Figure 4.2: Three vending machines

Nonetheless, strong bisimilarity does not deal with the relative frequency of different outcomes of the activities, thus this implies that external behaviour of the two systems may be distinguishable from external observer since some actions occur in different frequencies among the two systems.
In order to cope with this problem, the so-called *strong equivalence* is widely used, and the latter notion is based on *conditional transition rate*.

4.2 Strong Equivalence

Before introducing the notion of strong equivalence, we should clarify what is the *conditional transition rate*, and subsequently, *total conditional transition rate*, both definitions are those presented in [35, 38].

Definition 4.2.1. (Conditional Transition Rate) *Given two PEPA components, the* conditional transition rate *between two components P_i and P_j through a some action type α, is*

$$q(P_i, P_j, \alpha) = \sum_{(\alpha, r_\alpha) \in Act(P_i|P_j)} r_\alpha$$

where $Act(P_i|P_j) = \{| \ (\alpha, r_\alpha) \in Act(P_i) \mid P_i \xrightarrow{(\alpha, r_\alpha)} P_j \ |\}$.

The conditional transition is representing the rate at which a system proceeding its execution as component P_i advances in order to behave as component P_j by performing an activity of action type α. The non-diagonal elements of the infinitesimal generator matrix of the Markov process $\boldsymbol{Q}$ are formed by the transition rates $q(P_i, P_j)$. Instead, the elements belonging to the main diagonal are the negative sum of the off-diagonal elements of each row. In particular, it will be used the following notations: $q(P_i) = \sum_{j \neq i} q(P_i, P_j)$ and $q_{ii} = -q(P_i)$. For any finite and irreducible PEPA model P, the steady-state distribution $\Pi(\cdot)$ exists and it is possible to find it by solving the system of equation with normalization condition and the global balance equations:

$$\sum_{P_i \in ds(P)} \Pi(P_i) = 1$$

$$\Pi \boldsymbol{Q} = \boldsymbol{0}.$$

Hence, it is possible to define another quantity, the total conditional transition rate, which is based on the previous definition of conditional transition rate.

Definition 4.2.2. (Total Conditional Transition Rate) *Let S be a set of possible derivatives. The* total conditional transition rate *from P_i to S, denoted $q[P_i, S, \alpha]$ is defined by:*

$$q[P_i, S, \alpha] = \sum_{P_J \in S} q(P_i, P_j, \alpha)$$

.

Now, we can report the notion of *strong equivalence* is defined in [35]:

Definition 4.2.3. *Let us consider an equivalence relation $\mathcal{R}$, such that $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$. $\mathcal{R}$ is a* strong equivalence *if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$,*

$$q[P, S, \alpha] = q[Q, S, \alpha]$$

Relying on the definition above, two PEPA components are considered to be *strongly equivalent* in case for any action type α, the total conditional transition rates from those components to any equivalence class, formed by some equivalence relation between them, are equal. Moreover, we are interested also in the relation which is the largest strong equivalence, formed by the union of all strong equivalences. Consider the next definition:

Definition 4.2.4. (Strong Equivalence) *Two PEPA components P and Q are considered* strongly equivalent, *written $P \cong Q$, if $(P, Q) \in \mathcal{R}$ for some strong equivalence $\mathcal{R}$, i.e.,*

$$\cong \; = \bigcup \{\mathcal{R} \mid \mathcal{R} \textit{ is a strong equivalence}\}.$$

$\cong$ is called strong equivalence and it is the largest strong equivalence over PEPA components.

Furthermore, in [35] it is proved that $\cong$ is a congruence for PEPA by showing that the $\cong$ relation is preserved by the combinators of PEPA. Indeed, if $P_1 \cong P_2$ then

1. $a.P_1 \cong a.P_2$;
2. $P_1 + Q \cong P_2 + Q$;
3. $P_1 \underset{L}{\bowtie} Q \cong P_2 \underset{L}{\bowtie} Q$;
4. $P_1/L \cong P_2/L$.

The proofs are omitted, since they can be found in the Chapter 8 of [35].

4.3 Lumpable Bisimulation

In this section we introduce a bisimulation-like relation, called *lumpable bisimulation*, which has been introduced and analysed in [38].

The idea is that two PEPA components are considered *lumpably bisimilar* if for any action type $\alpha \neq \tau$, the total conditional transition rates from the considered component to any equivalence class, built from some equivalence relation between the components, are the same. In the following lines the formal definition is reported, taken from the above-mentioned paper.

Definition 4.3.1. (Lumpable Bisimulation) *Let us consider the equivalence relation over PEPA components $\mathcal{R}$ such that $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$.* Then *$\mathcal{R}$ is a* lumpable bisimulation *if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q, \notin \mathcal{S}$,*

it holds $q[P, \mathcal{S}, \alpha] = q[Q, \mathcal{S}, \alpha]$ where $q[\cdot]$ is the total conditional transition rate, introduced in the previous section, Definition 4.2.2.

As like as the strong equivalence case, the identity relation is trivially a lumpable bisimulation. Moreover, we also present some of propositions about *lumpable bisimulation*, introduced by the authors of [38].

Proposition 5. *Consider I, the set of indices and $\mathcal{R}_i$ a lumpable bisimulation relation for all $i \in I$. Then the transitive closure of their union, $\mathcal{R} = (\cup_{i \in I} \mathcal{R}_i)*$, results to be a lumpable bisimulation, too.*

In another words, the above proposition states that any union of lumpable bisimulations will produce a lumpable bisimulation.

Definition 4.3.2. (Lumpable Bisimilarity) *Two PEPA components P and Q are considered* lumpably bisimilar, *denoted as $P \approx_l Q$, if $(P, Q) \in \mathcal{R}$ for some lumpable bisimulation $\mathcal{R}$, meaning that,*

$$\approx_l \; = \bigcup \{\mathcal{R} \mid \mathcal{R} \textit{ is a lumpable bisimulation}\}.$$

$\approx_l$ is called lumpable bisimilarity and it is the largest symmetric lumpable bisimulation over PEPA components.

Notice that also lumpable bisimilarity, as like as the strong equivalence case, is a congruence for evaluation contexts.

Proposition 6. *If* $P_1 \approx_l P_2$ *then*

- *for all* $L \subseteq \mathcal{A}$, $P_1 \underset{L}{\bowtie} Q \approx_l P_2 \underset{L}{\bowtie} Q$;
- $P_1/L \approx_l P_2/L$.

The proof is omitted since is analogous to the items 3 and 4 in Proposition 8.3.1 of [35].

Moreover, the following proposition states that lumpable bisimilarity is a lumpable relation.

Proposition 7. *For all PEPA components* P, Q *such that* $P \approx_l Q$ *and for all* $S \in C/\approx_l$ *such that* $P, Q \in S$, $q(P, S) = q(Q, S)$.

The omitted proof is reported in the Section 4 of [38].

Now one might ask the difference between strong equivalence and lumpable bisimulation, since their definitions seem to be very similar to each other. However, note that the definition of strong equivalence previously presented is stricter than that of lumpable bisimulation. Indeed, lumpable bisimulation allows arbitrary activities with type τ among components belonging to the same equivalent class. Moreover, another difference is that lumpable bisimulation is a congruence with respect to the cooperation and hiding combinators but not for choice and the prefix operators. Taking also into account propositions presented previously, we can say that in general the lumpable bisimulation induces a grosser lumping than the strong equivalence but, it has stricter congruence properties.

In the following lines, practical examples of lumpable bisimulation are presented, in which the classical example of client-server model is used, where the server consists in a provider of some resource, the client is the requester of the resource.

Example 1. Let us consider the *client-server* model represented in the following pictures. The considered components are C for the *client*, S for the *server*.

The PEPA equations for the server model are the following:

$$\begin{aligned} S_{Think} &\stackrel{\text{def}}{=} (\tau, \delta).S_{Compute} \\ S_{Compute} &\stackrel{\text{def}}{=} (comp, \epsilon).S_{Send} + (\tau, \phi).S_{Error} \\ S_{Send} &\stackrel{\text{def}}{=} (tr, \eta).S_{Think} \\ S_{Error} &\stackrel{\text{def}}{=} (\tau, \zeta).S_{Recovery} \\ S_{Recovery} &\stackrel{\text{def}}{=} (\tau, \delta).S_{Compute} \end{aligned}$$

and the corresponding derivation graph is the following:

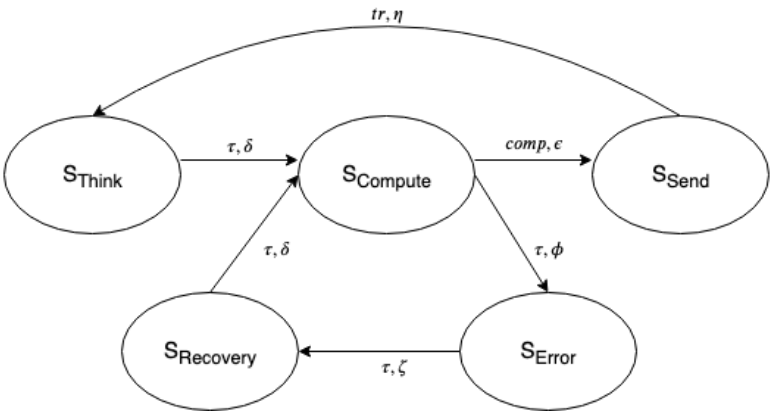

Figure 4.3: Server model

The PEPA equations for the client model are the following:

$$\begin{aligned} C_{Empty} &\stackrel{\text{def}}{=} (tr, \top).C_1 \\ C_i &\stackrel{\text{def}}{=} (\top, i\mu).C_{i+1} + (tr, \top).C_{i+1} + (send, \gamma).C_{Wait} \\ C_N &\stackrel{\text{def}}{=} (tr, \top).C_N + (send, \gamma).C_{Wait} \\ C_{Wait} &\stackrel{\text{def}}{=} (\tau, \nu).C_{Empty} \end{aligned}$$

for $1 \leq i < N$. The corresponding derivation graph is:

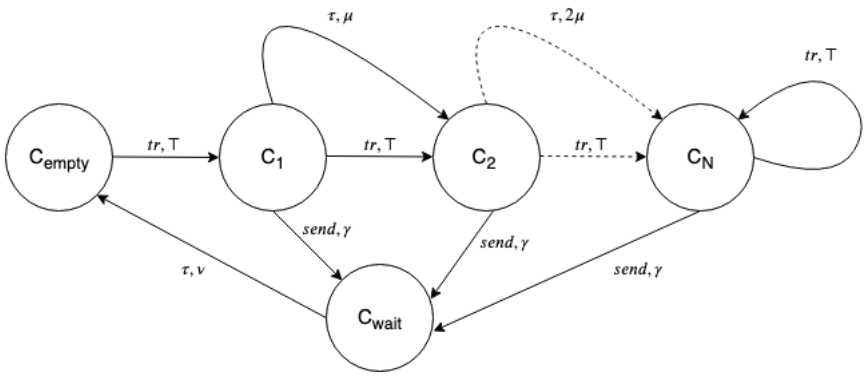

Figure 4.4: Client model

If we consider the server component we notice that $S_{Think} \approx_l S_{Recovery}$ and thus, they belong to the same equivalence class, we call this class $[S']$. The lumped server component is shown in the following Figure 4.5.

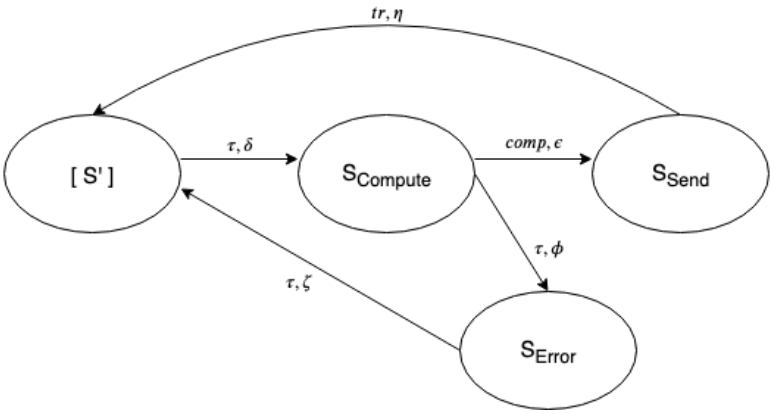

Figure 4.5: Lumping of the server model

If we consider the client, we observe that $C_1 \approx_l C_2 \approx_l \cdots \approx_l C_N$ and let $[C']$ be the associated equivalence class. The figure here below shows the lumped client model.

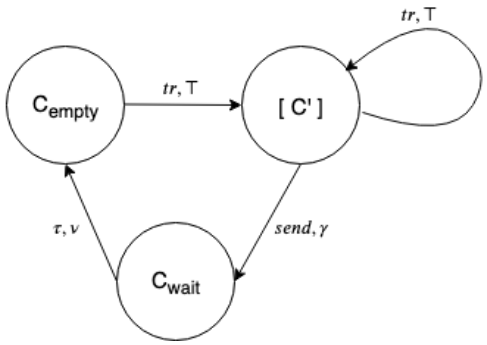

Figure 4.6: Lumping of the client model

Notice that, it can be said that S_{Think} is strongly equivalent to $S_{Recovery}$, however C_i is not strongly equivalent to C_j, with $i \neq j$. Indeed, the sum of the rate of the action with type τ outgoing from term C_i is $i\mu$ for $1 \leq i \leq N$, and is 0 if $i = N$, meaning that different for each term $C_1, \cdots, C_N$. Therefore, the joint model obtained by applying lumpable bisimilarity results to have less states than that obtained by strong equivalence.

Chapter 5

Quasi-lumpable Bisimulation and Proportional Bisimulation

In the previous chapter we have seen that lumpable bisimilarity is very similar to strong equivalence, since in order to consider two PEPA components equivalent they both require that the total conditional transition rates from these two components to any equivalence class must be the same. Although lumpable bisimulation is going to induce grosser lumping than the strong equivalence, since it allows arbitrary activities with type τ among components belonging to the same equivalent class, is still requiring strict conditions on the rates of components belonging to the system taken into account.
In this chapter, two types of bisimulation-like relation, named *quasi-lumpable bisimilarity* and *proportional bisimilarity*, which both extend the notion of *lumpable bisimilarity* for PEPA models in the attempt of relaxing the strict conditions, are presented.

5.1 Quasi-lumpable Bisimulation

Quasi-lumpable bisimulation is an equivalence relation based on the notion of quasi-lumpability presented in the Section 2.3.1 in Chapter 2. Recall that quasi-lumpability is founded on the idea that some small perturbations on the transition rates of Markov chain can be applied in order to make it lumpable. In particular, two PEPA components are *quasi-lumpable bisimilar with respect to* ϵ with $\epsilon \geq 0$ if there is an equivalence relation between them such that, for any action type α different from τ, the total conditional transition rates from

those components to any equivalence class, via activities of this type, are equal after a small perturbation of the system.

Definition 5.1.1. (Quasi-lumpable bisimulation) *An equivalence relation over PEPA components* $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, *is a* quasi-lumpable bisimulation with respect to ϵ with $\epsilon \geq 0$, *if whenever* $(P, Q) \in \mathcal{R}$ *then for all* $\alpha \in \mathcal{A}$ *and for all* $S \in \mathcal{C}/\mathcal{R}$,

- *either* $\alpha \neq \tau$,
- *or* $\alpha = \tau$ *and* $P, Q \notin S$,

it holds

$$\mid q[P, S, \alpha] - q[Q, S, \alpha] \mid \leq \epsilon,\ \epsilon \geq 0.$$

Henceforth, we would like to denote $P \approx_l^\epsilon Q$ if P and Q are quasi-lumpable bisimilar. Additionally, notice that quasi-lumpable bisimulation over the state space of a PEPA component P induces a quasi-lumpability on the state space of the Markov chain underlying P.

We must specify that this definition is similar to the notion of *approximate strong equivalence* introduced by the authors Dimitrios Milios and Stephen Gilmore in [45]. Nevertheless, quasi-lumpable bisimulation is less strict than that of approximate strong equivalence because quasi-lumpable bisimulation allows arbitrary activities with type τ among components belonging to the same equivalence class. In general, a quasi-lumpable bisimulation induces a grosser aggregation than the approximate strong equivalence of [45].
Nevertheless, we would like to avoid the use of the latter equivalence relation, the quasi-lumpable bisimulation for several reasons: first of all, it is really hard to build a testing algorithm in order to verify experimentally the partitioning of PEPA components according to the quasi-lumpable bisimulation relation. Notice that in [45] a partitioning strategy for PEPA components that minimizes an upper bound for approximate strong equivalence has been proposed. But this algorithm involves the use of clustering techniques making the algorithm itself complicated. The other reason is that quasi-lumpable bisimulation is not preserved by combinators of PEPA language, in particular it is not preserved by the cooperation combinator which represents the synchronization between components.

5.2 Proportional Bisimulation

In this section, the notion of proportional bisimulation is presented. Indeed, as specified in the previous section, unfortunately the notion of quasi-lumpable bisimulation with respect to a specific bound $\epsilon \geq 0$ is not preserved under union, meaning that the union of two quasi-lumpable bisimulations with respect to ϵ is still a quasi-lumpable bisimulation but, in general, not with respect to the same bound ϵ. In this context, the notion of proportional bisimulation has been ideated. Proportional bisimulation is characterized by a function κ that associates a real value κ_P to each PEPA component P. Then, consider the following definition:

Definition 5.2.1. (Proportional Bisimulation) *Let κ be a function from PEPA components to $\mathbb{R}^+$. An equivalence relation over PEPA components $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, is a* proportional bisimulation with respect to κ *if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$,*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q \notin S$,*

it holds

$$\frac{q[P, S, \alpha]}{\kappa_P} = \frac{q[Q, S, \alpha]}{\kappa_Q}.$$

Clearly, the identity relation is a proportional bisimulation for any function κ. As like as the strong equivalence and lumpable bisimulation case, we are interested in the relation which is the largest κ-proportional bisimulation, formed by the union of all κ-proportional bisimulation. Nevertheless, it is quite hard to see that this will indeed be a lumpable bisimulation.

Consider the following proposition which states that any union of κ-proportional bisimulations generates a κ-proportional bisimulation:

Proposition 8. *Let each $\mathcal{R}_i$, $i \in I$ for some index set I, be a κ-proportional bisimulation. Then $R = (\bigcup_{i \in I} \mathcal{R}_i)^*$, the transitive closure of their union, is also a κ-proportional bisimulation.*

Proof. $\mathcal{R}_i$ is an equivalence relation, consequently it can be stated that $\mathcal{R}$ is also an equivalence relation.
Let $\mathcal{C}/\mathcal{R}$ and $\mathcal{C}/\mathcal{R}_i$ denote the sets of equivalence classes, meaning that the

set of components $\mathcal{C}$ has been partitioned by $\mathcal{R}$ and each $\mathcal{R}_i$ respectively. By definition, $(P,Q) \in \mathcal{R}_i$ implies that $(P,Q) \in \mathcal{R}$, and so any equivalence class $S^i_j \in \mathcal{C}/\mathcal{R}_i$ is wholly contained within some equivalence class $T_k \in \mathcal{C}/\mathcal{R}$. Moreover, it follows that there is some set J^i_k such that $T_k = \bigcup\limits_{j \in J^i_k} S^i_j$.

We want to show that $\mathcal{R}$ satisfies Definition 5.2.1 by induction over n. Therefore, consider $(P,Q) \in \mathcal{R}$, then $(P,Q) \in (\bigcup_{i\in I} \mathcal{R}_i)^n$ for some $n > 0$. By $\mathcal{R}_n$ we indicate $(\bigcup_{i\in I} \mathcal{R}_i)^n$. For some $T_k \in \mathcal{C}/\mathcal{R}$ and any $\alpha \in \mathcal{A}$, let us take into account the total conditional transition rates from P and Q into T_k given that $(P,Q) \in \mathcal{R}_n$.
If $n = 1$, $(P,Q) \in \mathcal{R}_1$ implies that $(P,Q) \in \mathcal{R}_i$ for any $i \in I$, and by following the previous lines,

$$\frac{q[P,T_k,\alpha]}{\kappa_P} = \sum_{j\in J^i_k} \frac{q[P,S^i_j,\alpha]}{\kappa_P} = \sum_{j\in J^i_k} \frac{q[Q,S^i_j,\alpha]}{\kappa_Q} = \frac{q[Q,T_k,\alpha]}{\kappa_Q}.$$

In case $n > 1$, for all $\mathcal{R}_m$, where $m < n$, it will be assumed that whether $(P,Q) \in \mathcal{R}_m$ then,

$$\frac{q[P,T_k,\alpha]}{\kappa_P} = \frac{q[Q,T_k,\alpha]}{\kappa_Q}$$

$(P,Q) \in \mathcal{R}_n$ implies that $(P,Q) \in \mathcal{R}_i;\mathcal{R}_{n-1}$, meaning that there exists a component $C \in \mathcal{C}$ such that $(P,C) \in \mathcal{R}_i$ for some $i \in I$ and $(C,Q) \in \mathcal{R}_{n-1}$. Consequently, we can state that

$$\frac{q[P,T_k,\alpha]}{\kappa_P} = \frac{q[C,T_k,\alpha]}{\kappa_C}$$

and by the induction hypothesis,

$$\frac{q[C,T_k,\alpha]}{\kappa_C} = \frac{q[Q,T_k,\alpha]}{\kappa_Q}.$$

Therefore, we obtain

$$\frac{q[P,T_k,\alpha]}{\kappa_P} = \frac{q[Q,T_k,\alpha]}{\kappa_Q}$$

as required.
Hence, $\mathcal{R}$ is a κ-proportional bisimulation. □

The previous Proposition 8 allows one to define the maximal κ-proportional bisimulation as the union of all κ-proportional bisimulations.

Definition 5.2.2. (Proportional bisimilarity) *Let κ be a function from PEPA components to $\mathbb{R}^+$. Two PEPA components P and Q are κ-proportionally bisimilar, written $P \approx_l^\kappa Q$, if $(P,Q) \in \mathcal{R}$ for some κ-proportional bisimulation $\mathcal{R}$, i.e.*

$$\approx_l^\kappa \;=\; \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a } \kappa\text{-proportional bisimulation}\}.$$

$\approx_l^\kappa$ is called κ-proportional bisimilarity *and it is the largest symmetric κ-proportional bisimulation over PEPA components.*

It is possible also to define a weaker relation, *proportional bisimulation up to* $\approx_l^\kappa$: for the purpose of showing proportional bisimulation between two PEPA components we can seek for a proportional bisimulation up to $\approx_l^\kappa$ between them. This result will be presented in the Proposition 9. Now consider the next definition:

Definition 5.2.3. *$\mathcal{R}$ is a proportional bisimulation relation up to $\approx_l^\kappa$ if $\mathcal{R}$ is an equivalence relation over $\mathcal{C}$ and $(P,Q) \in \mathcal{R}$ implies that for all $\alpha \in \mathcal{A}$, and for all $T \in \mathcal{C}/(\approx_l^\kappa \mathcal{R} \approx_l^\kappa)$,*

$$\frac{q[P,T,\alpha]}{\kappa_P} = \frac{q[Q,T,\alpha]}{\kappa_Q}.$$

Formally, $(P,Q) \in \approx_l^\kappa \mathcal{R} \approx_l^\kappa$ if there exist P_1 and Q_1 such that $P \approx_l^\kappa P_1$, $P_1 \mathcal{R} Q_1$ and $Q_1 \approx_l^\kappa Q$. For all $P \in \mathcal{C}$, let S_P denote the equivalence class in $\mathcal{C}/\approx_l^\kappa$ which contains P, R_P the corresponding equivalence class in $\mathcal{C}/\mathcal{R}$ and T_P the corresponding equivalence class in $\mathcal{C}/(\approx_l^\kappa \mathcal{R} \approx_l^\kappa)$. Then we can see that

$$T_P = \{Q | P \approx_l^\kappa R \approx_l^\kappa Q\} = \bigcup \{S_{Q_1} | Q_1 \in R_{P_1} | P_1 \in S_P\}$$

It follows that any $T_P \in \mathcal{C}/(\approx_l^\kappa \mathcal{R} \approx_l^\kappa)$ is a union of equivalence classes $S_{Q_1} \in \mathcal{C}/\approx_l^\kappa$.

Then, consider the next lemma:

Lemma 3. *In case $\mathcal{R}$ is a proportional bisimulation up to $\approx_l^\kappa$, then the relation $\approx_l^\kappa \mathcal{R} \approx_l^\kappa$ results to be a proportional bisimulation.*

Proof. Let us consider two components P and Q such that $P \approx_\kappa \mathcal{R} \approx_l^\kappa Q$. Hence, there exist components P_1 and Q_1 such that $P \approx_l^\kappa P_1 \mathcal{R} Q_1 \approx_l^\kappa Q$. Furthermore, for all $S \in \mathcal{C}/\approx_l^\kappa$

$$\frac{q[P,S,\alpha]}{\kappa_P} = \frac{q[P_1,S,\alpha]}{\kappa_{P_1}},$$
$$\frac{q[Q_1,S,\alpha]}{\kappa_{Q_1}} = \frac{q[Q,S,\alpha]}{\kappa_Q}$$

and for all $T \in \mathcal{C}/(\approx_l^\kappa \mathcal{R} \approx_l^\kappa)$,

$$\frac{q[P_1,T,\alpha]}{\kappa_{P_1}} = \frac{q[Q_1,T,\alpha]}{\kappa_{Q_1}}.$$

We know that $T \in \mathcal{C}/(\approx_l^\kappa \mathcal{R} \approx_l^\kappa)$ is a union of $S \in \mathcal{C}/\approx_l^\kappa$, therefore for all such T we obtain,

$$\frac{q[P,T,\alpha]}{\kappa_P} = \frac{q[Q,T,\alpha]}{\kappa_Q}.$$

□

Finally, consider the next proposition:

Proposition 9. *If $\mathcal{R}$ is a proportional bisimulation up to $\approx_l^\kappa$ then $\mathcal{R} \subseteq \approx_l^\kappa$.*

The proof follows immediately from Lemma 3.

5.3 Proportional Bisimulation and Markov Processes

In this section we analyse the proportional bisimulation relation from the underlying Markov process viewpoint. In particular, we examine what we can derive about the corresponding Markov processes when $P \approx_l^\kappa Q$.
It is well known that the relation $\approx_l^\kappa$ partitions the set of components $\mathcal{C}$, and it is easy to see that, in case restricted to the derivative set of an arbitrary component P, the relation partitions this set. We indicate by $ds(P)/\approx_l^\kappa$ the set of equivalence classes built as explained in previous lines. Now consider the next proposition:

Proposition 10. *For any component P, $ds(P)/\approx_l^\kappa$ induces a κ-proportional partition on the state space of the Markov process corresponding to P.*

Proof. Let S_i and S_j denote arbitrary elements of $ds(P)/\approx_l^\kappa$, and consider any two elements of S_i, P_{ik} and P_{il}. Then since $P_{ij} \approx_l^\kappa P_{il}$,

$$\frac{q[P_{ik}, S_j]}{\kappa_{P_{ik}}} = \frac{q[P_{il}, S_j]}{\kappa_{P_{il}}}$$

where $\frac{q[P_{ik},S_j]}{\kappa_{P_{ik}}}$ is the unconditional probability of any activity by the component P_{ik} resulting in a derivative within the equivalence class S_j. Therefore, the partition $ds(P)/\approx_l^\kappa$ induces a κ-proportional lumpable partition on the state space of the Markov process underlying P.

From the Proposition 10, we can clearly state that if proportional bisimulation over the derivative set of a component is used to induce a partition of the state space of the Markov process, then the corresponding aggregation will result in a Markov process. Consequently, the aggregated process may be exploited to find the steady state distribution.

In the following lines, some examples of proportional lumpable PEPA models are presented.

Example 1. Consider a simple system depicted in Figure 5.1 which follows a Markov process and can be specified in PEPA as:

$$\begin{aligned}
D_0 &\stackrel{\text{def}}{=} (\alpha, 2r).D_1 \\
D_1 &\stackrel{\text{def}}{=} (\beta, s).D_2 \\
D_2 &\stackrel{\text{def}}{=} (\alpha, r).D_3 + (\alpha, r).D_1 \\
D_3 &\stackrel{\text{def}}{=} (\beta, s).D_0
\end{aligned}$$

The derivation graph for this system is shown in Figure 5.1.

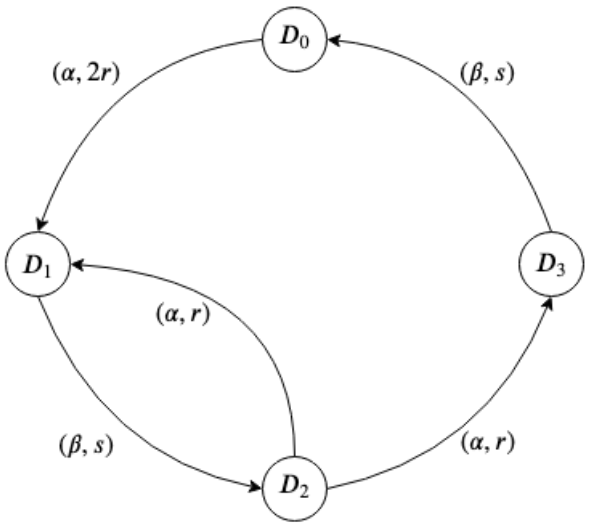

Figure 5.1: Derivation graph for the system described in *Example 1*

Let S be the set of possible derivatives. By definition of strong equivalence, we can easily see that $D_0 \cong D_2$ because $q[D_0, S, \alpha] = 2r$ and $q[D_2, S, \alpha] = r + r$. Similarly, $D_1 \cong D_3$ because $q[D_1, S, \beta] = s$ and $q[D_3, S, \beta] = s$.

Now we can modify the rates of the PEPA model depicted in Figure 5.1 in order to make the components not proportionally bisimilar. Differently from the previous example, one can notice that the rates of each activity is denoted by different Greek letters: in particular, in this example we have $r = 0.2$, $p = 0.2$, $t = 0.025$, $q = 0.0625$. Moreover, we consider the following factors κ_C for each component C present in the model: $\kappa_{D_0} = 4$, $\kappa_{D_1} = 2$, $\kappa_{D_2} = 8$, $\kappa_{D_3} = 5$. The new model with perturbed rates is depicted in Figure 5.2.
Therefore, we can see that D_0 is κ-proportionally bisimilar to D_2. Let again S be the set of possible derivatives:

$$\frac{q[D_0, S, \alpha]}{\kappa_{D_0}} = \frac{q[D_2, S, \alpha]}{\kappa_{D_2}}$$

because $\frac{q[D_0,S,\alpha]}{\kappa_{D_0}} = \frac{r}{4}$ and $\frac{q[D_2,S,\alpha]}{\kappa_{D_2}} = \frac{p+p}{8}$. By substituting the corresponding rates we have that $\frac{0.2}{4} = \frac{0.2+0.2}{8} = 0.05$.

Similarly, we can easily prove that $D_1 \approx_l^{\kappa} D_3$:

$$\frac{q[D_1, S, \beta]}{\kappa_{D_1}} = \frac{q[D_3, S, \beta]}{\kappa_{D_3}}$$

because $\frac{q[D_1,S,\beta]}{\kappa_{D_1}} = \frac{t}{2}$ and $\frac{q[D_3,S,\beta]}{\kappa_{D_3}} = \frac{q}{5}$. By substituting the corresponding rates we have that $\frac{0.025}{2} = \frac{0.0625}{5} = 0.0125$.

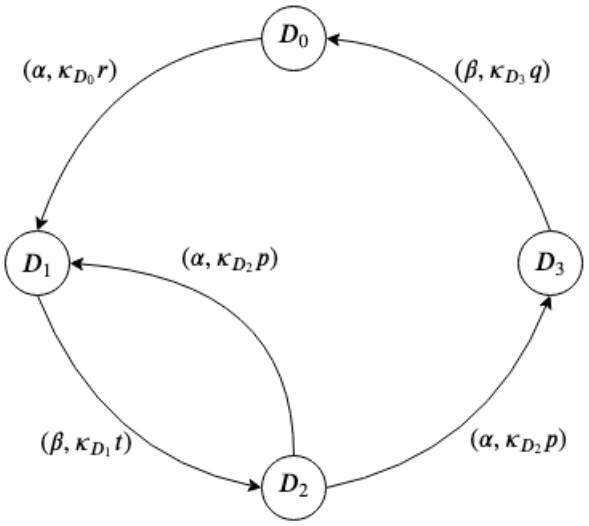

Figure 5.2: Modification on the rates of *Figure 5.1*

Example 2. Consider a simple buffer in which messages arrive according to a Poisson distribution with rate λ. The buffer is emptied and this follows an exponential distribution. The mean time between successive emptying is $n\mu^{-1}$ where n represents the number of items in the buffer. The buffer has capacity M and if it reaches the maximum capacity, arrival messages are lost. This buffer follows a Markov process and can be specified in PEPA as:

$$B_n = (\tau, \lambda).B_{n+1} \quad 0 \leq n \leq M-1$$

$$B_n = (cl, \mu n^{-1}).B_0 \quad 0 \leq n \leq M$$

The derivation graph for this system is shown in Figure 5.3 here below.

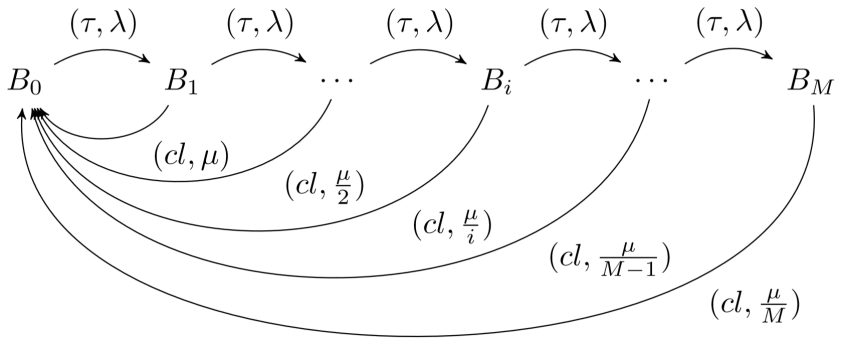

Figure 5.3: Original Buffer System

If we consider the function κ from PEPA components to $\mathbb{R}^+$ such that $\kappa_{B_0} = 1$ and $\kappa_{B_n} = 1/n$ for all n with $0 < n \leq M$, it is quite easy to see that $B_0 \approx_l^\kappa B_0^{'}$ where $B_0^{'}$ is depicted in Figure 5.4. From the steady-state equilibrium distribution of the reduced system we can then compute the steady-state equilibrium distribution of the original system.

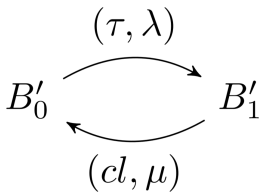

Figure 5.4: Reduced Buffer System

Chapter 6

Persistent Stochastic Non-Interference

6.1 Overview

In this chapter the notion of *Persistent Stochastic Non-Interference (PSNI)*, which has been introduced and studied in the namesake paper [37], is presented. Notice that all of the results and proof which are present in this chapter are all drawn from the article [37].

Persistent Stochastic Non-Interference is a *persistent* information flow security property for stochastic processes expressed as terms of the PEPA process algebra. For the last ten years, the study about the security of information systems has been considered a major issue for researchers. In the literature several security properties such as integrity, anonymity and confidentiality have been proposed and studied, but they totally ignore the time behaviour of the considered system. However, we must specify that from the timing observation of a system, attackers can infer some useful information for completing the attack (see, e.g., [9, 12, 21]). *Persistent Stochastic Non-Interference* tries to fill the gap between the two problems.

Before explaining the details about *Persistent Stochastic Non-Interference* we should clarify the two terms *stochastic* and *persistent* which characterize the PSNI property. The basic property on which *Persistent Stochastic Non-Interference* is founded is called *Non-Interference* security property which has

been introduced in [30] for deterministic systems. In particular, the development of non-interference for non-deterministic systems has been conducted in, e.g., [56]. Non-interference is defined in [37] as follows: "*Non-Interference is an information flow security property which aims at protecting sensitive data from undesired access.*" In particular, the interest is in protecting the high level information from the low level, unauthorized users: "*No information flow is possible from high to low if what is done at the high level cannot interfere in any way with the low level.*" It is well known that unwanted flows are not blocked by only adopting access control policies or cryptographic protocols. Indeed, non-interference property has been studied in several settings like cryptographic protocols [1, 14, 24], programming languages [26, 48, 51, 11], trace models [42, 44], process calculi [17, 18, 23, 34, 47, 10], timed models [25, 31] and stochastic models [3].
We call *Stochastic Non-Interference* property the *Non-Interference* property for stochastic, cooperating processes expressed as terms of PEPA. It has been inspired from the *Bisimulation based Non-Deducibility on Compositions (BNDC)* property for non-deterministic CCS processes introduced in [22] . *Stochastic Non-Interference* is a quantitative extension of the *Non-Interference* property. *Stochastic Non-Interference* becomes *Persistent Stochastic Non-Interference* when every state reachable by a process satisfies the *Stochastic Non-Interference* property, namely, the *Stochastic Non-Interference* property *persists* for every reachable state of a process taken into account.
In this chapter the notion of *Persistent Stochastic Non-Interference* is presented and studied in its details, by providing characterizations of the *PSNI* property by exploiting the structural operational semantics of PEPA.

6.2 Persistent Stochastic Non-Interference

Since *Persistent Stochastic Non-Interference* property is based on the basic property *Stochastic Non-Interference*, we should first of all define the latter property, in order to understand the *persistent* version of it. Notice that we report several definitions by following the steps done by the authors of [37].
Stochastic Non-Interference (SNI) property ensures the security of a process P in the following way: a process P is considered secure whether a low level observer is not able to discriminate the behaviour of P running in isolation from the behaviour of P in cooperation with other, high level process H. For the purpose of giving a formal definition of *Stochastic Non-Interference (SNI)*,

the set of visible action types $\mathcal{A} \setminus \{\tau\}$ is partitioned into two sets, $\mathscr{H}$ and $\mathscr{L}$ of high and low level action types, respectively. A high level component is usually denoted by H and it consists in a PEPA component such that for all $H' \in ds(H)$, $\mathcal{A}(H') \subseteq \mathscr{H}$, therefore, every derivative of H commit in only high level actions. The set of all high level PEPA components is denoted by $\mathcal{C}_H$.
Unfortunately, *Stochastic Non-Interference (SNI)* is a basic security property that is not guaranteed to be preserved during the whole system execution: this means that even if a system satisfies *Stochastic Non-Interference* property, the system can still reach unsafe states. In order to cope with this problem, the notion of *Persistent Stochastic Non-Interference (PSNI)* has been introduced in [37]. *Persistent Stochastic Non-Interference* necessitate that every reachable state of a system is considered secure, meaning that a process P is secure if and only if

$$\forall P' \textit{ reachable from } P, \textit{ P' satisfies SNI.}$$

More precisely, the authors of [37] state that the challenge of *Persistent Stochastic Non-Interference (PSNI)* consists in considering all of the possible information flow from a *classified (high)* level of confidentiality to an *untrusted (low)* one. A quite strict requirement of this definition is that any information flow is not allowed, even if a malicious processes run at the classified level. In particular, *Persistent Stochastic Non-Interference* can be used to protect a system from internal attacks, like *Trojan Horse* programs, which are malicious software that disguise to appear not suspicious but hide some malicious code inside them.

The notion of *Persistent Stochastic Non-Interference* concerns the check for all the states reachable by the system with respect to every possible high level potential interactions. In another words, a system P satisfies *Persistent Stochastic Non-Interference* property if for every state P' reachable from P and for every high level process H a low level user is not able to distinguish P' from $P' \underset{\mathscr{H}}{\bowtie} H$, therefore one cannot distinguish P' running in isolation or, equivalently, P' in parallel execution with any high level PEPA component that does not synchronize with it from $P' \underset{\mathscr{H}}{\bowtie} H$ where H is a high component cooperating with P'. This means that a system P satisfies *PSNI* if what a low level user sees of the system is not modified when it cooperates with any high level process H.

In order to model a process P in individual execution, the special component

0 is introduced which helps in extending the syntax of non-sequential PEPA component as follows:

$$P ::= \mathbf{0} \mid P \underset{L}{\bowtie} P \mid P/L \mid S,$$

where **0** is representing the component that does not perform any activity.
First of all a model of a process P running in isolation is needed. P can be equivalently considered as running in parallel with any high level PEPA component that does not synchronize with it as $(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})$.
Consequently, even when high level interactions are allowed, these are assumed not to be visible by an external low level observer, which can only recognize the delay while the interactions take place. Therefore, from the low level point of view, the process $(P \underset{\mathscr{H}}{\bowtie} H)$ behaves as $(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}$.
Hence, $(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})$ and $(P \underset{\mathscr{H}}{\bowtie} H)$ should be indistinguishable for a low level user, meaning that $(P \underset{\mathscr{H}}{\bowtie} H\mathbf{0})/\mathscr{H}$ and $(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}$ have the same behaviour.
Persistent Stochastic Non-Interference is based on the bisimulation-like equivalence relation, indeed, this kind of behavioural equivalence at the base of our definition relies on the notion of lumpable bisimilarity.

Now, all the necessary tools in order to formally define the properties have been introduced. Hence, consider the next definition of *Stochastic Non-Interference* property:

Definition 6.2.1. (Stochastic Non-Interference) *Let P be a PEPA component.*

$$P \in SNI \text{ iff } \forall H \in \mathcal{C}_H,$$

$$(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l (P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

The equivalence relation $\approx_l$ is the *lumpable bisimilarity*, see definition 4.3.1.

Equivalently, also the persistent version of *Stochastic Non-Interference* can be defined. Consider the following definition of *Persistent Stochastic Non-Interference* property:

Definition 6.2.2. (Persistent Stochastic Non-Interference) *Let P be a PEPA component.*

$$P \in PSNI \text{ iff } \forall P' \in ds(P), \forall H \in \mathcal{C}_H,$$

$$P' \in SNI, \text{ i.e., } (P' \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

Notice that henceforth, with $P \setminus \mathscr{H}$ we denote the PEPA component $(P' \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H}$. Intuitively $P \setminus \mathscr{H}$ denotes the component P prevented from performing high level actions.

Therefore, we can give an alternative characterization of *Persistent Stochastic Non-Interference* as follows:

Proposition 11. *Let P be a PEPA component.*

$$P \in PSNI \text{ iff } \forall P' \in ds(P), \forall H \in \mathcal{C}_H,$$

$$P' \in SNI, \text{ i.e., } P' \setminus \mathscr{H} \approx_l (P' \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H}.$$

Now, it is possible to prove that *Persistent Stochastic Non-Interference* is preserved by lumpable bisimulation relation. Consider the next lemma and its proof:

Lemma 4. *If P and Q are two PEPA components such that $P \approx_l Q$ and $P \in PSNI$ then also $Q \in PSNI$.*

Proof. One can easily recognize that for all $Q' \in ds(Q)$ there is $P' \in ds(P)$ such that $P' \approx_l Q'$. From $P' \approx_l Q'$ the fact that $P' \in PSNI$ is contextual and we have $Q' \setminus \mathscr{H} \approx_l P' \setminus \mathscr{H} \approx_l (P' \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H} \approx_l (Q' \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H}$, $\forall H \in \mathcal{C}_H$.□

Now, one should notice that this property exploits two universal quantifications: the first one is over all the reachable states; the other is contained within the definition of *Stochastic Non-Interference*, over all the possible high level processes that can possibly interact with the system taken into account. To cope with this problem, the authors of [37] have introduced a novel bisimulation based equivalence relation over PEPA components, named $\approx_l^{hc}$ that allows one to give a characterization of *PSNI* without quantification over all the high level components H. In layman's terms, two processes are $\approx_l^{hc}$-equivalent if they can simulate each other in any possible high context, specifically, every context $C[\_]$ of the form $(\_ \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H}$ where $H \in \mathcal{C}_H$. Notice that for any high context $C[\_]$ and PEPA model P, all the states reachable from $C[P]$ have the form $C'[P']$ with $C'[\_]$ being a high context too and $P' \in ds(P)$.
The concept of *lumpable bisimulation on high contexts* is based on the idea is that, given two PEPA models P and Q, when a high level context $C[\_]$ filled with P executes a certain activity engaging in a transition from P to P', then the same context filled with Q is able to reproduce this step moving Q to Q' so

that P' and Q' are again lumpably bisimilar on high context, and vice-versa. Notice that this must be true for every possible high context $C[\_]$. It is important to note that the quantification over all possible high contexts is reiterated for P' and Q'. For a PEPA model P, $\alpha \in \mathcal{A}$, $S \subseteq ds(P)$ and a high context $C[\_]$ we define:

$$q_C(P, P', \alpha) = \sum_{C[P] \xrightarrow{(\alpha, r_\alpha)} C'[P']} r_\alpha$$

and

$$q_C[P, S, \alpha] = \sum_{P' \in S} q_C(P, P', \alpha).$$

Consider the next definition of *lumpable bisimulation on high contexts*:

Definition 6.2.3. (Lumpable bisimilarity on high contexts) *An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is a* lumpable bisimulation on high context *if whenever $(P, Q) \in \mathcal{R}$ then for all high context $C[\_]$, for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q \notin S$,*

it holds

$$q_C[P, S, \alpha] = q_C[Q, S, \alpha].$$

Two PEPA components P and Q are lumpably bisimilar on high contexts, *written $P \approx_l^{hc} Q$, if $(P, Q) \in \mathcal{R}$ for some lumpable bisimulation on high context $\mathcal{R}$, i.e.,*

$$\approx_l^{hc} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a lumpable bisimulation on high contexts.}\}$$

$\approx_l^{hc}$ is called lumpable bisimilarity on high contexts *and it is the largest symmetric lumpable bisimulation on high context over PEPA components. It is easy to prove that $\approx_l^{hc}$ is an equivalence relation.*

The next theorem, Theorem 5, provides a characterization of *Persistent Stochastic Non-Interference* in terms of $\approx_l^{hc}$. Nevertheless, the relation $\approx_l^{hc}$ still contains a universal quantification over all high level contexts. However, the characterization of *PSNI* given in Theorem 5 provides another look to the meaning and the properties of *PSNI*. Notice that the theorem statement and its proof are those introduced by the authors of [37].

Theorem 5. *Let P be a PEPA component. Then*

$$P \in PSNI \text{ iff } P \setminus \mathscr{H} \approx_l^{hc} P.$$

Proof. First of all we should prove that $P \setminus \mathscr{H} \approx_l^{hc} P$ implies $P \in PSNI$. Consider the following relation:

$$\mathcal{R} = \{((P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}) \mid H \in \mathcal{C}_H \text{ and } P \approx_l^{hc} Q\}.$$

From the knowledge that $\approx_l^{hc}$ is an equivalence relation, it can be easily said that also $\mathcal{R}$ is an equivalence relation. For the purpose of proving that $P \in PSNI$ it is sufficient to show that $\mathcal{R}$ is a lumpable bisimulation. Indeed, if $P \setminus \mathscr{H} \approx_l^{hc} P$ then for all $P' \in ds(P)$ there is $P'' \setminus \mathscr{H} \in ds(P \setminus \mathscr{H})$ such that $P'' \setminus \mathscr{H} \approx_l^{hc} P'$ and, by definition of $\mathcal{R}$, for all $H \in \mathcal{C}_H$, $((P'' \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}) \in \mathcal{R}$. Since $\mathcal{R}$ is a lumpable bisimulation, it can be stated that for all $H \in \mathcal{C}_H$, $(P'' \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \approx_l P'' \setminus \mathscr{H} \approx_l (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}$. In particular, there can exists $\bar{H} \in \mathcal{C}_H$ such that $(P' \underset{\mathscr{H}}{\bowtie} \bar{H})/\mathscr{H}$ is coincident with $P' \setminus \mathscr{H}$. Since we know that $\approx_l$ is an equivalence relation, by symmetry and transitivity, for every $P' \in ds(P)$ and for every $H \in \mathcal{C}_H$, $P'' \setminus \mathscr{H} \approx_l P' \setminus \mathscr{H} \approx_l (P' \underset{\mathscr{H}}{\bowtie} \bar{H})/\mathscr{H}$, i.e., $P \in PSNI$.
Let $\alpha \in \mathcal{A}$. Now we should show that if $(P \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H), (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \in \mathcal{R}$ and $S \in \mathcal{C}/\mathcal{R}$ then either $\alpha \neq \tau$ and $q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha] = q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]$ or $\alpha = \tau$ and if $(P \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H), (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \notin S$ then $q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha] = q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]$. Notice that, by definition of $\mathcal{R}$, $S \in \mathcal{C}/\mathcal{R}$ if and only if there exists $S' \in \mathcal{C}/\approx_l^{hc}$ such that $S = \{(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \mid P \in S'\}$. The proof follows immediately from the fact that if $P \approx_l^{hc} Q$ then

- if $\alpha \neq \tau$ then $q_C[P, S', \alpha] = q_C[Q, S', \alpha]$ for all $S' \in \mathcal{C}/\approx_l^{hc}$ that is $q[(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, S, \alpha] = q[(Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, S, \alpha]$ for all $S \in \mathcal{C}/\mathcal{R}$.
- if $\alpha = \tau$ and $P, Q \notin S'$ then $q_C[P, S', \alpha] = q_C[Q, S', \alpha]$ for all $S' \in \mathcal{C}/\approx_l^{hc}$ that is $q[(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, S, \alpha] = q[(Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, S, \alpha]$ for all $S \in \mathcal{C}/\mathcal{R}$ with $(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \notin S$.

Now it is necessary to prove that if $P \in PSNI$ then $P \setminus \mathscr{H} \approx_l^{hc} P$. Let us consider the relation

$$\mathcal{R} = \{(P_1 \setminus \mathscr{H}, P_2) \mid P_1 \setminus \mathscr{H} \approx_l P_2 \setminus \mathscr{H} \text{ and } P_2 \in PSNI\}.$$

Let $\mathcal{R}^*$ be the reflexive, symmetric and transitive closure of $\mathcal{R}$. It is necessary to show $\mathcal{R}^*$ is a lumpable bisimulation on high contexts. $\mathcal{R}^*$ contains pairs of the following types:

- $(P_1 \setminus \mathscr{H}, P_2) \in \mathcal{R}$
- $(P_1 \setminus \mathscr{H}, P_3 \setminus \mathscr{H})$ such that $(P_1 \setminus \mathscr{H}, P_2) \in \mathcal{R}$ and $(P_3 \setminus \mathscr{H}, P_2) \in \mathcal{R}$, i.e., $P_2 \in PSNI$ and $P_1 \setminus \mathscr{H} \approx_l P_2 \setminus \mathscr{H} \approx_l P_3 \setminus \mathscr{H}$
- (P_1, P_3) such that $(P_2 \setminus \mathscr{H}, P_1) \in \mathcal{R}$ and $(P_2 \setminus \mathscr{H}, P_3) \in \mathcal{R}$, i.e., $P_1 \in PSNI$, $P_3 \in PSNI$ and $P_1 \setminus \mathscr{H} \approx_l P_2 \setminus \mathscr{H} \approx_l P_3 \setminus \mathscr{H}$
- all the symmetric pairs with respect to the above pairs and all the identity pairs

Notice that

$$\mathcal{R}^* = \{(P, Q) \mid P \in PSNI, Q \in PSNI \text{ and } P \setminus \mathscr{H} \approx_l Q \setminus \mathscr{H}\} \cup Id.$$

First of all, we show that for each equivalence class $S \in \mathcal{C}/\mathcal{R}^*$ with $|S|> 1$ there exist $S'_1, S'_2, ..., S'_k \in \mathcal{C}/\approx_l$ such that $S = \cup_{i=1}^{k} S'_i$. For the purpose of showing this it is sufficient that if $|S|> 1$ and $S' \cap S \neq \emptyset$ with $S' \in \mathcal{C}/\approx_l$ then $S' \subseteq S$. Since $|S|> 1$ and $S' \cap S \neq \emptyset$ there exists $P \in PSNI$ and $P \in S' \cap S$. Let $Q \in S'$ then $P \approx_l Q$ and hence by Lemma 4 $Q \in PSNI$ and $P \setminus \mathscr{H} \approx_l Q \setminus \mathscr{H}$, so $Q \in S$.

Consider a high context $C[\_]$ and some $\alpha \in \mathcal{A}$. It is necessary to prove if $(P, Q) \in \mathcal{R}^*$ and $S \in \mathcal{C}/\mathcal{R}^*$ then either $\alpha \neq \tau$ and $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$ or $\alpha = \tau$ and if $P, Q \notin S$ then $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$. We consider only the case $P \neq Q$ since the case $P = Q$ is obvious. This ensures that P and Q are $PSNI$.

- Assume $\alpha \neq \tau$. From the fact that $P, Q \in PSNI$ and $P \setminus \mathscr{H} \approx_l Q \setminus \mathscr{H}$ it follows that for all $S' \in \mathcal{C}/\approx_l$, $q_C[P, S', \alpha] = q_C[Q, S', \alpha]$. Since $S \in \mathcal{C}/\mathcal{R}^*$ is the union of classes of $\mathcal{C}/\approx_l$ then we have that $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$.
- Assume $\alpha = \tau$ and $P, S \notin S$. One can notice that, by definition of $\mathcal{R}^*$, also $P \setminus \mathscr{H}$, $Q \setminus \mathscr{H} \notin S$. Since $P, Q \in PSNI$ and $P \setminus \mathscr{H} \approx_l Q \setminus \mathscr{H}$ it follows that for all $S' \in \mathcal{C}/\approx_l$ such that $P \setminus \mathscr{H}$, $Q \setminus \mathscr{H} \notin S'$, $q_C[P, S', \tau] = q_C[Q, S', \tau]$. Since $S \in \mathcal{C}/\mathcal{R}^*$ is the union of classes of $\mathcal{C}/\approx_l$ then we have that $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$.

□

The authors of [37] have proved that it is possible to characterize the property *PSNI* by keeping away both the universal quantification over all the possible high level components and the universal quantification over all the possible reachable states. For the lumpable bisimulation case, we have noticed that it implicitly contains a quantification over all possible high contexts. Now the challenge is to express $\approx_l^{hc}$ in a rather simpler way by trying to use exclusively local information. This can be done by defining a novel equivalence relation which focuses only on observable equivalence where actions from $\mathscr{H}$ are ignored.

Consider the next definition of the notion of *lumpable bisimilarity up to* $\mathscr{H}$:

Definition 6.2.4. (Lumpable bisimilarity up to $\mathscr{H}$) *An equivalence relation over PEPA components,* $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ *is a* lumpable bisimulation up to $\mathscr{H}$ *if whenever* $(P, Q) \in \mathscr{R}$ *then for all* $\alpha \in \mathcal{A}$ *and for all* $S \in \mathcal{C}/\mathcal{R}$

- *if* $\alpha \notin \mathscr{H} \cup \{\tau\}$ *then*

$$q[P, S, \alpha] = q[Q, S, \alpha],$$

- *if* $\alpha \in \mathscr{H} \cup \{\tau\}$ *and* $P, Q, \notin S$ *then*

$$q[P, S, \alpha] = q[Q, S, \alpha].$$

Two PEPA components P *and* Q *are* lumpably bisimilar up to $\mathscr{H}$, *written* $P \approx_l^{\mathscr{H}} Q$, *if* $(P, Q) \in \mathcal{R}$ *for some lumpable bisimulation up to* $\mathscr{H}$, *i.e.,*

$$\approx_l^{\mathscr{H}} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a lumpable bisimulation up to } \mathscr{H}\}.$$

$\approx_l^{\mathscr{H}}$ *is called* lumpable bisimilarity up to $\mathscr{H}$ *and it is the largest symmetric lumpable bisimulation up to* $\mathscr{H}$ *over PEPA components. It is easy to prove that* $\approx_l^{hc}$ *and* $\approx_l^{\mathscr{H}}$ *are equivalent.*

The next theorem proves that relations $\approx_l^{hc}$ and $\approx_l^{\mathscr{H}}$ are equivalent. Consider the following lines:

Theorem 6. *Let* P *and* Q *be two PEPA components. Then*

$$P \approx_l^{hc} Q \text{ if and only if } P \approx_l^{\mathscr{H}} Q.$$

Proof. First of all, it will be shown that $P \approx_l^{hc} Q$ implies $P \approx_l^{\mathscr{H}}$. For this purpose we will show that $\approx_l^{hc}$ is a lumpable bisimulation up to $\mathscr{H}$. This follows from the following cases.

- Consider $\alpha \notin \mathscr{H} \cup \{\tau\}$. Since $P \approx_l^{hc} Q$ it holds that for all $S \in \mathcal{C}/\approx_l^{hc}$ and for all high context $C[\_]$, $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$. Since $\alpha \notin \mathscr{H} \cup \{\tau\}$, we have that $q[P, S, \alpha] = q[Q, S, \alpha]$.

- Let $\alpha \in \mathscr{H} \cup \{\tau\}$. From the fact that $P \approx_l^{hc} Q$ it holds that for all $S \in \mathcal{C}/\approx_l^{hc}$ such that $P, Q \notin S$ and for all high context $C[\_]$, $q_C[P, S, \tau] = q_C[Q, S, \tau]$. If $C[\_]$ does not synchronize neither with P nor with Q, we have that $q[P, S, \tau] = q[Q, S, \tau]$. On the other hand, let $C[\_]$ be a context with only one current action type $h \in \mathscr{H}$. Then, from $q_C[P, S, \tau] = q_C[Q, S, \tau]$ and $q[P, S, \tau] = q[Q, S, \tau]$, it follows that if P cooperates over h then also Q cooperates over h and $q[P, S, h] = q[Q, S, h]$.

Now, it is possible to show if $P \approx_l^{\mathscr{H}} Q$ then $P \approx_l^{hc} Q$. For this purpose is sufficient to prove that $\approx_l^{\mathscr{H}}$ is a lumpable bisimulation on high contexts. This follows from the following cases.

- Let $\alpha \notin \mathscr{H} \cup \{\tau\}$. From the fact that $P \approx_l^{\mathscr{H}} Q$ it holds that for all $S \in \mathcal{C}/\approx_l^{\mathscr{H}}$, $q[P, S, \alpha] = q[Q, S, \alpha]$. Since a high context can only perform high level activities, we have that $q[P, S, \alpha] = q_C[P, S, \alpha]$ and $q[Q, S, \alpha] = q_C[Q, S, \alpha]$ for all high context $C[\_]$. Hence, $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$.

- Let $\alpha = \tau$. We prove that $q_C[P, S, \alpha] = q_C[Q, S, \alpha]$ for all high level context $C[\_]$. Henceforward, the inductive technique on the number of current action types of $C[\_]$ that synchronize with P and Q will be used. Since $P \approx_l^{\mathscr{H}} Q$ it holds that for all $S \in \mathcal{C}/\approx_l^{\mathscr{H}}$ such that $P, Q \notin S$, $q[P, S, \tau] = q[Q, S, \tau]$. If $C[\_]$ does not synchronize with P and Q we have that $q[P, S, \tau] = q_C[P, S, \tau]$ and $q[Q, S, \tau] = q_C[Q, S, \tau]$, i.e., $q_C[P, S, \tau] = q_C[Q, S, \tau]$. Let $C[\_]$ be a context that has only one current action type $h \in \mathscr{H}$ that synchronizes with P and Q. From the fact that $P \approx_l^{\mathscr{H}} Q$ it holds that for all $S \in \mathcal{C}/\approx_l^{\mathscr{H}}$ such that $P, Q \notin S$, $q[P, S, h] = q[Q, S, h]$ and $q[P, S, \tau] = q[Q, S, \tau]$, we get $q_C[P, S, \tau] = q_C[Q, S, \tau]$. The inductive step trivially follows.

□

Theorem 6 allows us to identify a local property of processes, with no quantification on the states and on the high contexts, which is a necessary and

sufficient condition for *Persistent Stochastic Non-Interference*. Consider the following corollary:

Corollary 6.1. *Let P be a PEPA component. Then*

$$P \in PSNI \text{ iff } P \setminus \mathscr{H} \approx_l^{\mathscr{H}} P.$$

In the end, we are able to provide a characterization of *PSNI* in terms of *unwinding conditions*. Specifically, if a state P' of a *PSNI* PEPA model P execute a high level activity leading it to a state P'', then P' and P'' are not distinguishable for a low level observer. Then consider the next theorem:

Theorem 7. *Let P be a PEPA component.*

$$P \in PSNI \text{ iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \text{ implies } P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}.$$

Proof. First of all, it will be proved that if $P \in PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$. Indeed, by Proposition 12, $P' \in PSNI$ and thus, by Corollary 6.1, $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P'$. By Definition 6.0.4 of $\approx_l^{\mathscr{H}}$, for all $S \in \mathcal{C}/\approx_l^{\mathscr{H}}$ such that $P' \setminus \mathscr{H}$, $P' \notin S$, both $q[P' \setminus \mathscr{H}, S, \tau] = q[P', S, \tau]$ and $q[P' \setminus \mathscr{H}, S, h] = q[P'S, h]$. Since $P' \setminus \mathscr{H}$ does not perform any high level action, $q[P' \setminus \mathscr{H}, S, h] = 0$ while, since $P' \xrightarrow{(h,r)} P''$, $q[P', S, h] \neq 0$. Therefore, from $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P'$, either h is not a current action type of P' or $P' \setminus \mathscr{H}$, $P' \in S$, i.e., $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P''$. Since also $P'' \in PSNI$, from $P'' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P''$ it follows that $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P'' \setminus \mathscr{H}$. Finally, since both $P' \setminus \mathscr{H}$ and $P'' \setminus \mathscr{H}$ do not perform any high level activity, $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P'' \setminus \mathscr{H}$ is equivalent to $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$.

Now it is possible to show that if for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$ then $P \in PSNI$. In particular, by Corollary 6.1 we have to prove that $P \setminus \mathscr{H} \approx_l^{\mathscr{H}} P$. Let $\mathcal{R} = \{(P' \setminus \mathscr{H}, P'') \mid P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}\}$. If we prove that $\mathcal{R}$ is a lumpable bisimilarity up to $\mathscr{H}$, then we have the thesis. Indeed, one can observe that from $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$, for all $\alpha \notin \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ we get $q[P' \setminus \mathscr{H}, S, \alpha] = q[P'', S, \alpha]$. Finally, if $\alpha \in \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ we have $q[P' \setminus \mathscr{H}, S, \alpha] = 0$ and if $P' \xrightarrow{(h,r)} P''$ then $P' \setminus \mathscr{H}$, $P'' \in S$. Hence $\mathcal{R}$ is a lumpable bisimilarity up to $\mathscr{H}$. □

Now, by exploiting the relation $\approx_l^{\mathscr{H}}$, *Persistent Stochastic Non-Interference* property can also be defined as follows.

Theorem 8. *Let P be a PEPA component.*

$$P \in PSNI \text{ iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \text{ implies } P' \approx_l^{\mathscr{H}} P''.$$

Proof. First of all, one can show that if $P \in PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \approx_l^{\mathscr{H}} P''$. Indeed, by Proposition 12, $P' \in PSNI$ and therefore, by Corollary 6.1, $P' \setminus \mathscr{H} \approx_l^{\mathscr{H}} P'$. By Definition 6.0.4 of $\approx_l^{\mathscr{H}}$, for all $S \in \mathcal{C}/\approx_l^{\mathscr{H}}$ such that $P' \setminus \mathscr{H}$, $P' \notin S$, both $q[P' \setminus \mathscr{H}, S, \tau] = q[P', S, \tau]$ and $q[P' \setminus \mathscr{H}, S, h] = q[P'S, h]$. Hence, if $P' \setminus \mathscr{H}$, $P' \notin S$, since $P' \setminus \mathscr{H}$ is not able to perform any high level transition it holds $q[P' \setminus \mathscr{H}, S, h] = q[P', S, h] = 0$. Let S'' be the equivalence class of P''. Since $P' \xrightarrow{(h,r)} P''$ it holds $q[P', S'', h] \neq 0$ and it has to be $P' \in S''$.

Now it is possible to prove that if for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \approx_l^{\mathscr{H}} P''$, then $P \in PSNI$. In particular, by Corollary 6.1 we have to prove that $P \setminus \mathscr{H} \approx_l^{\mathscr{H}} P$. Let $\mathcal{R} = \{(P' \setminus \mathscr{H}, P') \mid P' \in ds(P)\} \cup \approx_l^{\mathscr{H}}$. If we show that $\mathcal{R}$ is a lumpable bisimilarity up to $\mathscr{H}$, then we have the thesis. Indeed, one can observe that for all $\alpha \notin \mathscr{H}$ and for all P'' it holds $q(P' \setminus \mathscr{H}, P'' \setminus \mathscr{H}, \alpha) = q(P', P'', \alpha)$, so for all $S \in \mathcal{C}/\mathcal{R}$ we get $q[P' \setminus \mathscr{H}, S, \alpha] = q[P', S, \alpha]$. Finally, if $\alpha \in \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ we have $q[P' \setminus \mathscr{H}, S, \alpha] = 0$. Since $\approx_l^{\mathscr{H}} \subseteq \mathcal{R}$ by the hypothesis we have that if $P' \notin S$, then $q[P', S, \alpha] = 0$. Hence, $\mathcal{R}$ is a lumpable bisimilarity up to $\mathscr{H}$. □

Corollary 6.1 and Theorems 7 and 8 provide different characterizations of *PSNI* and they naturally allow to apply different efficient methods for the verification and construction of secure systems.

6.3 Properties of Persistent Stochastic Non-Interference

In this section, some interesting properties of *PSNI* studied in [37] are reported. In particular, it is proved that *PSNI* is compositional with respect to low prefix, cooperation over low actions and hiding.

Proposition 12. *Let P and Q be two PEPA components. If $P, Q \in PSNI$ then*

- $(\alpha, r).P \in PSNI$ *for all* $\alpha \in \mathscr{L} \cup \{\tau\}$
- $P/L \in PSNI$ *for all* $L \subseteq \mathcal{A}$
- $P \underset{L}{\bowtie} Q \in PSNI$ *for all* $L \subseteq \mathscr{L}$

Proof. Let us assume that $P, Q \in PSNI$.

- If $P \in PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$. This property is naturally maintained for the PEPA component $(\alpha, r).P$ when $\alpha \in \mathscr{L} \cup \{\tau\}$.

- If $P \in PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$. Let $L \subseteq \mathcal{A}$ and $P'/L \in ds(P)$. Assume that $P'/L \xrightarrow{(h,r)} P''/L$. Since $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$, subsequently $(P' \underset{\mathscr{H}}{\bowtie} \bar{H}) \approx_l (P'' \underset{\mathscr{H}}{\bowtie} \bar{H})$ for any high level PEPA component $\bar{H}$ that does not cooperate with P. Since lumpable bisimilarity is a congruence for the evaluation contexts, for all $L \subseteq \mathcal{A}$, $(P' \underset{\mathscr{H}}{\bowtie} \bar{H})/L \approx_l (P'' \underset{\mathscr{H}}{\bowtie} \bar{H})/L$. One can assume that $\vec{\mathcal{A}}(\bar{H}) \cap L = \emptyset$ and therefore, since also $\vec{\mathcal{A}}(\bar{H}) \cap \vec{\mathcal{A}}(\bar{P}) = \emptyset$, $(P'/L \underset{\mathscr{H}}{\bowtie} \bar{H})/L \approx_l (P''/L \underset{\mathscr{H}}{\bowtie} \bar{H})/L$, i.e., $(P'/L) \setminus \mathscr{H} \approx_l (P''/L) \setminus \mathscr{H}$.

- If $P, Q \in PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$ and for all $Q' \in ds(Q)$, $Q' \xrightarrow{(h,r)} Q''$ implies $Q' \setminus \mathscr{H} \approx_l Q'' \setminus \mathscr{H}$. Let $L \subseteq \mathscr{L}$ and $P' \underset{L}{\bowtie} Q' \in ds(P \underset{L}{\bowtie} Q)$. Consider that $P' \underset{L}{\bowtie} Q' \xrightarrow{(h,r)} P'' \underset{L}{\bowtie} Q''$. In this case, either $P' \xrightarrow{(h,r)} P''$ or $Q' \xrightarrow{(h,r)} Q''$. Let us assume that $P' \xrightarrow{(h,r)} P''$ and then $(P' \underset{L}{\bowtie} Q') \xrightarrow{(h,r)} (P'' \underset{L}{\bowtie} Q')$. Since the hypothesis is $P \in PSNI$, we have that $P' \setminus \mathscr{H} \approx_l P'' \setminus \mathscr{H}$, i.e., $(P' \underset{\mathscr{H}}{\bowtie} \bar{H}) \approx_l (P'' \underset{\mathscr{H}}{\bowtie} \bar{H})$ for any high level PEPA component $\bar{H}$ that does not cooperate with P and Q. From the fact that $\approx_l$ is a congruence with respect to the cooperation operator we have $(P' \underset{\mathscr{H}}{\bowtie} \bar{H}) \underset{L}{\bowtie} (Q' \underset{\mathscr{H}}{\bowtie} \bar{H}) \approx_l (P'' \underset{\mathscr{H}}{\bowtie} \bar{H}) \underset{L}{\bowtie} (Q' \underset{\mathscr{H}}{\bowtie} \bar{H})$, moreover since $\mathscr{H} \cap L = \emptyset$ we obtain $(P' \underset{L}{\bowtie} Q') \underset{\mathscr{H}}{\bowtie} \bar{H} \approx_l (P'' \underset{L}{\bowtie} Q') \underset{\mathscr{H}}{\bowtie} \bar{H}$, i.e., $(P' \underset{L}{\bowtie} Q') \setminus \mathscr{H} \approx_l (P'' \underset{L}{\bowtie} Q') \setminus \mathscr{H}$. In case that $Q' \xrightarrow{(h,r)} Q''$ the proof is analogous.

□

The fact that *PSNI* is not preserved by the choice operator is a consequence of the fact that lumpable bisimilarity is not a congruence for this operator.

Example 1. In this example we will consider a web server which is able to serve the requests coming from clients. First of all, the client makes a request through the activity *req* with rate ρ. Consequently, the server can directly reply with type *res* and rate μ or it can redirect the request to another server, for example in a shopping web site, there can exists a redirect to a payment service. The server which receives the redirection is going to process a high level authentication with type *log* and rate λ and then a reply is sent to the client with type *res* and rate μ. From the payment service viewpoint, the reply sent from P_3 to P_1 may denote the confirmation of the payment received by the customer. We will show that the system presented in this example satisfies *PSNI*, but it can be useful to understand the guidelines suggested by the theoretical property in order to design system models. In particular, it is important to notice that the observer should not to be able to understand if a payment took place. There are two suggestions given by *PSNI* to achieve this. First, the reply sent by P_3 must have the same rate of that sent by P_2. If this is not the case, the malicious observer may statistically infer what the customer is doing. The second suggestion consists in the following: once the system is in P_2, there must be a race policy between *res* and *log* even if the system intends to perform a *log* activity. If this is not the case, the malicious observer would be able to statistically detect the high level from the convolution of two exponential random variables with rates λ and μ. This may be implemented by sending from P_2 fictitious messages to P_1 with the aim of confusing the observer.
Formally, the system has the following *PEPA* specification:

$$P_1 \stackrel{\text{def}}{=} (res, \rho).P_2$$
$$P_2 \stackrel{\text{def}}{=} (res, \mu).P_1$$
$$P_2 \stackrel{\text{def}}{=} (log, \lambda).P_3$$
$$P_3 \stackrel{\text{def}}{=} (res, \mu).P_1$$

and its derivation graph is depicted in Figure 6.1 here below.

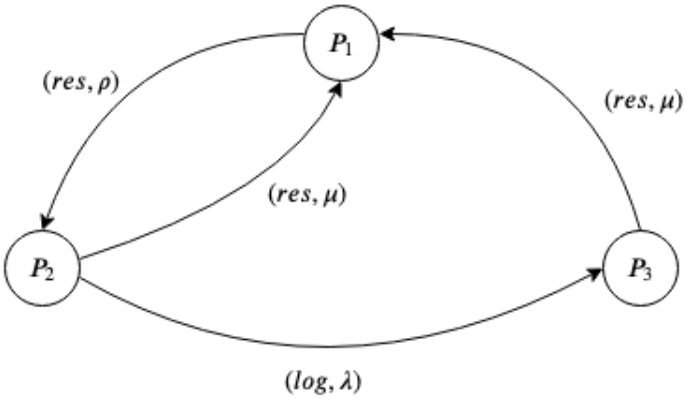

Figure 6.1: A simple three state model

In this case, following Theorem 7, we can prove that $P_1 \in PSNI$. Indeed, it is easy to prove that $P_2 \setminus \mathscr{H} \approx_l P_3 \setminus \mathscr{H}$ when $\approx_l$ is the lumpable bisimilarity, meaning that P_2 and P_3 are not distinguishable for a low level observer. In particular, the probability for a low level user to observe, in steady-state, the system being in state P_1 doesn't depend on the behaviour of P_2. Namely, there is no importance of how frequently P_2 performs high level activity (log, λ). To see this, suppose that P_2 always synchronizes on *log*. Then for a low level observer, the system behaves as $P_1/\mathscr{H}$ as depicted in Figure 6.2 here below.

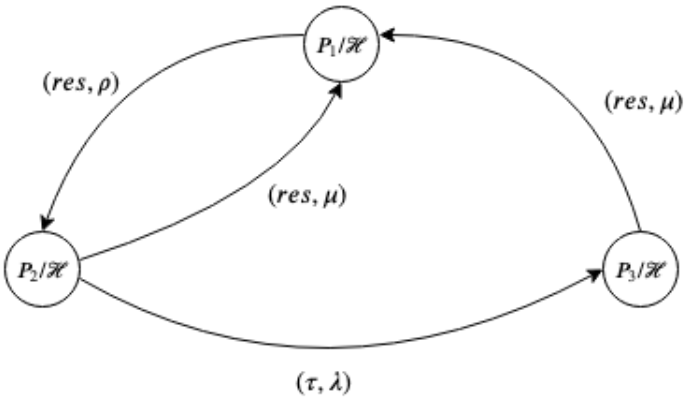

Figure 6.2: The model of $P_1/\mathscr{H}$

We can compute the steady-state distribution of $P_1/\mathscr{H}$ by solving the

global balance equations together with the normalization condition, obtaining:

$$\begin{aligned} \pi\rho &= \pi_2\mu + \pi_3\mu \\ \pi_2(\lambda+\mu) &= \pi_1\rho \\ \pi_3\mu &= \pi_2\lambda \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

whose solution is

$$\begin{aligned} \pi_1 &= \frac{\mu}{\mu+\rho} \\ \pi_2 &= \frac{\mu\rho}{(\mu+\rho)(\lambda+\mu)} \\ \pi_3 &= \frac{\lambda\rho}{(\mu+\rho)(\lambda+\mu)} \end{aligned}$$

where π_1, π_2 and π_3 denote the steady-state probabilities of states $P_1/\mathscr{H}$, $P_2/\mathscr{H}$ and $P_3/\mathscr{H}$, respectively.

Consider now the case in which P_2 never synchronizes over *log*. Then, the low level view of the system is represented by $P_1 \setminus \mathscr{H}$ depicted in Figure 6.3 here below.

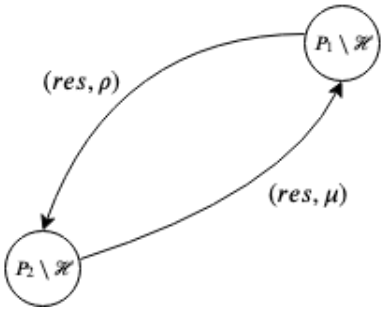

Figure 6.3: The model of $P_1 \setminus \mathscr{H}$

Again, we can compute the steady-state distribution of $P_1 \setminus \mathscr{H}$ by solving the global balance equations with the normalization condition, obtaining:

$$\begin{aligned} \pi'\rho &= \pi'_2\mu \\ \pi'_2(\lambda+\mu) &= \pi'_1\rho \\ \pi'_1 + \pi'_2 &= 1 \end{aligned}$$

whose solution is

$$\begin{aligned} \pi'_1 &= \frac{\mu}{\mu+\rho} \\ \pi'_2 &= \frac{\rho}{\mu+\rho} \end{aligned}$$

where π'_1 and π'_2 are the steady-state probabilities of $P_1 \setminus \mathscr{H}$ and $P_2 \setminus \mathscr{H}$, respectively. This shows that, from the low level point of view, the steady-state probability of P_1 is independent of the fact that P_2 has cooperated with a high level context or not.

Chapter 7

Approximate Persistent Stochastic Non-Interference

7.1 Overview

In the previous chapter, we have seen that the information security flow property *Persistent Stochastic Non-Interference* is one of the very few results, within the security of information field of research, which takes into account the timing behaviour of the analysed system. Indeed, from the user's point of view, it is of crucial importance to be ensured that a system is secure and he can really trust it: the system should guarantee for all its users that sensitive information never flows to unauthorized entities and in this context *Persistent Stochastic Non-Interference* property will result very useful for systems specified as terms of the quantitative Markovian process algebra, PEPA language.
PSNI is based on bisimulation-like equivalence relation inducing a lumping on the underlying Markov chain: in the formal definition of *PSNI* given in Definition 6.2.2 in Chapter 6, we can see that the equivalence relation lumpable bisimulation $\approx_l$ is used. More specifically, for a low level user $(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})$ and $(P \underset{\mathscr{H}}{\bowtie} H)$ are indistinguishable, meaning that $(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H}$ and $(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}$ are lumpably bisimilar. Lumpable bisimulation, introduced in Definition 4.3.1 in Chapter 4, requires that for all activity in the activity set $\mathcal{A}$ and for all equivalence class $S \in \mathcal{C}/\mathcal{R}$, the total conditional transition rate from one component to any equivalence class is the same to the total conditional transition rate from another component to the same equivalence class. It is going to equate all the reachable and indistinguishable states of the system and the probability

of observing the system in a state P is the sum of probabilities of the states lumpably bisimilar to P. Nevertheless, this requirement results to be too much restrictive, since it is very difficult to find a real world system which satisfies the lumpability condition for the underlying Markov chain, and consequently very few real system models will satisfy *PSNI* based on lumpable bisimulation. For the purpose of coping this problem, in this chapter the notions of *Quasi-Persistent Stochastic Non Interference* and *Approximate-Persistent Stochastic Non Interference* are presented. They consist in the attempts of relaxing the requirements of the property *PSNI*, by exploiting equivalence relations such as *quasi-lumpable bisimulation* and *proportional bisimulation* introduced and studied in previous sections.
This chapter is divided in two main sections: in the first section the *quasi-Persistent Stochastic Non-Interference* based on quasi-lumpable bisimulation is introduced and studied by analysing its problems, while in the second section *approximate-Persistent Stochastic Non-Interference* based on proportional bisimulation is presented and studied in its details.

7.2 Quasi-Persistent Stochastic Non-Interference

In this section the notion of *quasi-PSNI* is presented. As the name suggests, it is based on the equivalence relation *quasi-lumpable bisimulation* $\approx_l^\epsilon$ with respect a bound ϵ, with $\epsilon \geq 0$, which has been introduced in Chapter 5. Henceforward, *quasi-Persistent Stochastic Non Interference* property will be indicated shortened as *Q-PSNI*.

In order to formally define *Q-PSNI* we should first of all define *Q-SNI*, as in the previous *PSNI* case. Therefore, consider the next definition, presented in the same way as *SNI* definition in Definition 6.2.1 in the previous chapter:

Definition 7.2.1. (Quasi-Stochastic Non Interference) *Let P be a PEPA component and ϵ be a bound, $\epsilon \geq 0$.*

$$P \in \textit{Q-SNI iff } \forall H \in \mathcal{C}_H,$$

$$(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l^\epsilon (P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

Now, also the persistent version of *Q-SNI* can be defined. Consider the following definition:

Definition 7.2.2. (Quasi-Persistent Stochastic Non-Interference) *Let P be a PEPA component and ϵ be a bound, $\epsilon \geq 0$.*

$$P \in \textit{Q-PSNI iff } \forall P' \in ds(P),\ \forall H \in \mathcal{C}_H,$$

$$P' \in \textit{Q-SNI, i.e., } (P' \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l^{\epsilon} (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

However, it may result dangerous to involve the equivalence relation quasi-lumpable bisimulation within the characterization of *PSNI*. Indeed, the main problem we may encounter with the notion of quasi-lumpable bisimulation is that, the latter equivalence relation with respect to a specific bound $\epsilon \geq 0$ is not preserved under union, meaning that the union of two quasi-lumpable bisimulations with respect to ϵ is a quasi-lumpable bisimulation, but in general not with respect to the same bound. For this reason, a construction of secure systems by adopting *Q-PSNI* may engage in a very complicated task, so that, we may prefer the use of proportional bisimulation equivalence relation in order to newly define the *PSNI* property in a safer way.

7.3 Approximate Persistent Stochastic Non-Interference

We have just seen in the previous section that *Q-PSNI* is not the optimal alternative to the existing *PSNI* property, since *Q-PSNI* is based on quasi-lumpable bisimulation which is not preserved under the union. In this context, we can still try to relax the property *PSNI* by using another equivalence relation, *proportional bisimulation*, denoted as $\approx_l^{\kappa}$, which has been previously introduced in Chapter 5. Henceforward, we indicate the approximate-Persistent Stochastic Non Interference as *A-PSNI*.

In order to formally define *A-PSNI* we should first of all define *A-SNI*, as in the previous *PSNI* case. Therefore, consider the next definition, presented in the same way as *SNI* definition in Definition 6.2.1 in the previous chapter:

Definition 7.3.1. (Approximate-Stochastic Non-Interference) *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$.*

$$P \in \textit{A-SNI iff } \forall H \in \mathcal{C}_H,$$

$$(P \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l^\kappa (P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

Now, we can define the persistent version of *A-SNI*. Consider the next definition:

Definition 7.3.2. (Approximate-Persistent Stochastic Non-Interference) *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$.*

$$P \in \textit{A-PSNI iff } \forall P' \in ds(P),\ \forall H \in \mathcal{C}_H,$$

$$P' \in \textit{A-SNI, i.e., } (P' \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H} \approx_l^\kappa (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

Henceforth, with $P \setminus \mathscr{H}$ we denote the PEPA component $(P' \underset{\mathscr{H}}{\bowtie} \mathbf{0})/\mathscr{H}$. Intuitively $P \setminus \mathscr{H}$ denotes the component P prevented from performing high level actions. Therefore, we can characterize the *A-PSNI* property as follows:

Proposition 13. *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$.*

$$P \in \textit{A-PSNI iff } \forall P' \in ds(P),\ \forall H \in \mathcal{C}_H,$$

$$P' \setminus \mathscr{H} \approx_l^\kappa (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}.$$

Now that we have both the definition of *A-SNI* and *A-PSNI*, we can try to present further properties and characterization of *A-PSNI*. Consider the next lemma:

Lemma 9. *Let P and Q be PEPA component, H a high level PEPA component and κ a function from PEPA components to $\mathbb{R}^+$ such that $\kappa_{P \underset{\mathscr{H}}{\bowtie} H} = \kappa_P$ for all PEPA component P and for all high level component H. If P and Q are two PEPA components such that $P \approx_l^\kappa Q$ then $(P \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H} \approx_l^\kappa (Q \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H}$.*

The lemma here above is going to state that if two components are proportionally bisimilar, then the cooperation of one component with any high level component, prevented from performing any high level activity is equivalent to the cooperation of the other component with the same high level component, prevented from performing any high level activity, too. The latter result will be useful in order to prove theorems which will be presented in this section.

Non we are able to prove that *A-PSNI* is preserved by proportional bisimulation. Consider the next lemma:

Lemma 10. *Let P and Q be PEPA component and κ a function from PEPA components to $\mathbb{R}^+$. If P and Q are two PEPA components such that $P \approx_l^\kappa Q$ and $P \in$ A-PSNI then also $Q \in$ A-PSNI.*

Proof. First of all, one can notice that for all $Q' \in ds(Q)$ there exists $P' \in ds(P)$ such that $P' \approx_l^\kappa Q'$. From $P' \approx_l^\kappa Q'$ the fact that $P' \in$ *A-PSNI* can derived from Lemma 9. Additionally, from Lemma 10, we have $Q' \setminus \mathscr{H} \approx_l^\kappa P' \setminus \mathscr{H} \approx_l^\kappa (P' \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H} \approx_l^\kappa (Q' \underset{\mathscr{H}}{\bowtie} H) \setminus \mathscr{H}$, $\forall H \in \mathcal{C}_H$. □

As in the *PSNI* case, we want to characterize *A-PSNI* without quantification over all the high level components H. In order to achieve this, we should define a new equivalence relation called *proportional bisimulatiton on high context*. In particular, consider the following lines:

Definition 7.3.3. (Proportional bisimilarity on high contexts) *Let κ be a function from PEPA components to $\mathbb{R}^+$. An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is a* proportional bisimulation on high context *if whenever $(P, Q) \in \mathcal{R}$ then for all high context $C[\_]$, for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q \notin S$,*

it holds

$$\frac{q_C[P, S, \alpha]}{\kappa_P} = \frac{q_C[Q, S, \alpha]}{\kappa_Q}.$$

Two PEPA components P and Q are proportionally bisimilar on high contexts, *written $P \approx_{l,\kappa}^{hc} Q$, if $(P, Q) \in \mathcal{R}$ for some proportional bisimulation on high context $\mathcal{R}$, i.e.,*

$$\approx_{l,\kappa}^{hc} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a proportional bisimulation on high contexts.}\}$$

$\approx^{hc}_{l,\kappa}$ *is called* proportional bisimilarity on high contexts *and it is the largest symmetric proportional bisimulation on high context over PEPA components. Moreover,* $\approx^{hc}_{l,\kappa}$ *is an equivalence relation.*

Theorem 11 gives a characterization of *A-PSNI* in terms of $\approx^{hc}_{l,\kappa}$, even if $\approx^{hc}_{l,\kappa}$ still contains a universal quantification over all high level contexts. Consider the following theorem:

Theorem 11. *Let* P *be a PEPA component and* κ *a function from PEPA components to* $\mathbb{R}^+$*. Then*

$$P \in \textit{A-PSNI iff } P \setminus \mathscr{H} \approx^{hc}_{l,\kappa} P.$$

Proof. First of all, it is proved that $P \setminus \mathscr{H} \approx^{hc}_{l,\kappa} P$ implies $P \in$ *A-PSNI*. Consider the following relation

$$\mathcal{R} = \{((P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}) \mid H \in \mathcal{C}_H \textit{ and } P \approx^{hc}_{l,\kappa} Q\}.$$

Since $\approx^{hc}_{l,\kappa}$ is an equivalence relation, it follows that also $\mathcal{R}$ is an equivalence relation. To prove that $P \in$ *A-PSNI* it is sufficient to show that $\mathcal{R}$ is a proportional bisimulation. In fact, if $P \setminus \mathscr{H} \approx^{hc}_{l,\kappa} P$ then for all $P' \in ds(P)$ there exists $P'' \setminus \mathscr{H} \in ds(P \setminus \mathscr{H})$ such that $P'' \setminus \mathscr{H} \approx^{hc}_{l,\kappa} P'$ and, by definition of $\mathcal{R}$, for all $H \in \mathcal{C}_H$, $((P'' \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}) \in \mathcal{R}$. Since $\mathcal{R}$ is a proportional bisimulation, we have that for all $H \in \mathcal{C}_H$,
$(P'' \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H} \approx^{\kappa}_{l} (P' \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}$. In particular, there exists $\bar{H} \in \mathcal{C}_H$ such that $(P' \underset{\mathscr{H}}{\bowtie} \bar{H})/\mathscr{H}$ coincides with $P' \setminus \mathscr{H}$. Since $\approx^{\kappa}_{l}$ is an equivalence relation, by symmetry and transitivity, therefore for every $P' \in ds(P)$ and for every $H \in \mathcal{C}_H$, $P'' \setminus \mathscr{H} \approx^{\kappa}_{l} P' \setminus \mathscr{H} \approx^{\kappa}_{l} (P' \underset{\mathscr{H}}{\bowtie} \bar{H})/\mathscr{H}$, i.e., $P \in$ *A-PSNI*.
Consider $\alpha \in \mathcal{A}$. One should prove that if $((P \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H), (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}) \in \mathcal{R}$ and $S \in \mathcal{C}/\mathcal{R}$ then either $\alpha \neq \tau$ and $\frac{q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]}{\kappa_{P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}} = \frac{q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]}{\kappa_{Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}}$ or $\alpha = \tau$ and if $(P \setminus \mathscr{H} \underset{\mathscr{H}}{\bowtie} H), (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \notin S$ then $\frac{q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]}{\kappa_{P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}} = \frac{q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}, S, \alpha]}{\kappa_{Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}}$.
One should recognize that, by definition of $\mathcal{R}$, $S \in \mathcal{C}/\mathcal{R}$ if and only if there exists $S' \in \mathcal{C}/\approx^{hc}_{l,\kappa}$ such that $S = \{(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \mid P \in S'\}$. The proof follows immediately from the fact that if $P \approx^{hc}_{l,\kappa} Q$ then

- if $\alpha \neq \tau$ then $\frac{q_C[P,S',\alpha]}{\kappa_P} = \frac{q_C[Q,S',\alpha]}{\kappa_Q}$ for all $S' \in \mathcal{C}/\approx^{hc}_{l,\kappa}$ that is $\frac{q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H},S,\alpha]}{\kappa_{P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}} = \frac{q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H},S,\alpha]}{\kappa_{Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}}$ for all $S \in \mathcal{C}/\mathcal{R}$.
- if $\alpha = \tau$ and $P, Q \notin S'$ then $\frac{q_C[P,S',\alpha]}{\kappa_P} = \frac{q_C[Q,S',\alpha]}{\kappa_Q}$ for all $S' \in \mathcal{C}/\approx^{hc}_{l,\kappa}$ that is $\frac{q[P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H},S,\alpha]}{\kappa_{P \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}} = \frac{q[Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H},S,\alpha]}{\kappa_{Q \underset{\mathscr{H}}{\bowtie} H/\mathscr{H}}}$ for all $S \in \mathcal{C}/\mathcal{R}$ with $(P \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H}, (Q \underset{\mathscr{H}}{\bowtie} H)/\mathscr{H} \notin S$.

Now it is possible to show if $P \in$ *A-PSNI* then $P \setminus \mathscr{H} \approx^{hc}_{l,\kappa} P$. Let us consider the relation

$$\mathcal{R} = \{(P_1 \setminus \mathscr{H}, P_2) \mid P_1 \setminus \mathscr{H} \approx^{\kappa}_{l} P_2 \setminus \mathscr{H} \textit{ and } P_2 \in \textit{approximate-PSNI}\}.$$

Let $\mathcal{R}^*$ be the reflexive, symmetric and transitive closure of $\mathcal{R}$. Then one should prove that $\mathcal{R}^*$ is a proportional bisimulation on high contexts. $\mathcal{R}^*$ contains pairs of the following types:

- $(P_1 \setminus \mathscr{H}, P_2) \in \mathcal{R}$
- $(P_1 \setminus \mathscr{H}, P_3 \setminus \mathscr{H})$ such that $(P_1 \setminus \mathscr{H}, P_2) \in \mathcal{R}$ and $(P_3 \setminus \mathscr{H}, P_2) \in \mathcal{R}$, i.e., $P_2 \in$ *A-PSNI* and $P_1 \setminus \mathscr{H} \approx^{\kappa}_{l} P_2 \setminus \mathscr{H} \approx^{\kappa}_{l} P_3 \setminus \mathscr{H}$
- (P_1, P_3) such that $(P_2 \setminus \mathscr{H}, P_1) \in \mathcal{R}$ and $(P_2 \setminus \mathscr{H}, P_3) \in \mathcal{R}$, i.e., $P_1 \in$ *A-PSNI*, $P_3 \in$ *A-PSNI* and $P_1 \setminus \mathscr{H} \approx_{l} P_2 \setminus \mathscr{H} \approx^{\kappa}_{l} P_3 \setminus \mathscr{H}$
- all the symmetric pairs with respect the above pairs and all the identity pairs

Notice that

$$\mathcal{R}^* = \{(P, Q) \mid P \in \textit{A-PSNI}, Q \in \textit{A-PSNI and } P \setminus \mathscr{H} \approx^{\kappa}_{l} Q \setminus \mathscr{H}\} \cup \textit{Id}.$$

First of all it is necessary to prove that for each equivalence class $S \in \mathcal{C}/\mathcal{R}^*$ with $|S| > 1$ there exist $S'_1, S'_2, ..., S'_k \in \mathcal{C}/\approx^{\kappa}_{l}$ such that $S = \cup_{i=1}^{k} S'_i$. For the purpose of proving this it is sufficient that if $|S| > 1$ and $S' \cap S \neq \emptyset$ with $S' \in \mathcal{C}/\approx^{\kappa}_{l}$ then $S' \subseteq S$. Since $|S| > 1$ and $S' \cap S \neq \emptyset$ there exists $P \in$ *A-PSNI* and $P \in S' \cap S$. Let $Q \in S'$ then $P \approx^{\kappa}_{l} Q$ and hence by Lemma 11 $Q \in$ *A-PSNI* and $P \setminus \mathscr{H} \approx^{\kappa}_{l} Q \setminus \mathscr{H}$, so $Q \in S$.

Let $C[\_]$ be a high context and $\alpha \in \mathcal{A}$. We have to prove that if $(P,Q) \in \mathcal{R}^*$ and $S \in \mathcal{C}/\mathcal{R}^*$ then either $\alpha \neq \tau$ and $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$ or $\alpha = \tau$ and if $P,Q \notin S$ then $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$. We consider only the case $P \neq Q$ since the case $P = Q$ is trivial. This ensures that P and Q are *A-PSNI*.

- Let us assume $\alpha \neq \tau$. From the fact that $P,Q \in$ *A-PSNI* and $P \setminus \mathscr{H} \approx_l^\kappa Q \setminus \mathscr{H}$ follows that for all $S' \in \mathcal{C}/\approx_l^\kappa$, $\frac{q_C[P,S',\alpha]}{\kappa_P} = \frac{q_C[Q,S',\alpha]}{\kappa_Q}$. Since $S \in \mathcal{C}/\mathcal{R}^*$ is the union of classes of $\mathcal{C}/\approx_l^\kappa$ then we have that $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$.
- Assume $\alpha = \tau$ and $P,S \notin S$. One should notice that, by definition of $\mathcal{R}^*$, also $P \setminus \mathscr{H}$, $Q \setminus \mathscr{H} \notin S$. Since $P,Q \in$ *A-PSNI* and $P \setminus \mathscr{H} \approx_l^\kappa Q \setminus \mathscr{H}$ it follows that for all $S' \in \mathcal{C}/\approx_l^\kappa$ such that $P \setminus \mathscr{H}$, $Q \setminus \mathscr{H} \notin S'$, $\frac{q_C[P,S',\tau]}{\kappa_P} = \frac{q_C[Q,S',\tau]}{\kappa_Q}$. Since $S \in \mathcal{C}/\mathcal{R}^*$ is the union of classes of $\mathcal{C}/\approx_l^\kappa$ then we have that $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$.

□

As in the previous section, let us define a novel equivalence relation which focuses only on observable equivalence where actions from $\mathscr{H}$ may be ignored.

Again, as in the *PSNI* case, we want to show how it is possible to give a characterization of *A-PSNI* avoiding both the universal quantification over all the possible high level components and the universal quantification over all the possible reachable states. Therefore, consider the next notion of *proportional bisimilarity up to* $\mathscr{H}$:

Definition 7.3.4. (Proportional bisimilarity up to $\mathscr{H}$) *Let κ be a function from PEPA components to $\mathbb{R}^+$. An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is a* proportional bisimulation up to $\mathscr{H}$ *if whenever $(P,Q) \in \mathscr{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$*

- *if $\alpha \notin \mathscr{H} \cup \{\tau\}$ then*

$$\frac{q[P,S,\alpha]}{\kappa_P} = \frac{q[Q,S,\alpha]}{\kappa_Q},$$

- *if $\alpha \in \mathscr{H} \cup \{\tau\}$ and $P,Q, \notin S$ then*

$$\frac{q[P,S,\alpha]}{\kappa_P} = \frac{q[Q,S,\alpha]}{\kappa_Q}.$$

Two PEPA components P and Q are proportionally bisimilar up to $\mathscr{H}$, *written $P \approx^{\mathscr{H}}_{l,\kappa} Q$, if $(P,Q) \in \mathcal{R}$ for some proportional bisimulation up to $\mathscr{H}$, i.e.,*

$$\approx^{\mathscr{H}}_{l,\kappa} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a proportional bisimulation up to } \mathscr{H}\}.$$

$\approx^{\mathscr{H}}_{l,\kappa}$ is called proportional bisimilarity up to $\mathscr{H}$ *and it is the largest symmetric proportional bisimulation up to $\mathscr{H}$ over PEPA components. It is easy to prove that $\approx^{hc}_{l,\kappa}$ and $\approx^{\mathscr{H}}_{l,\kappa}$ are equivalent.*

Consider the next theorem in order to see that $\approx^{hc}_{l,\kappa}$ and $\approx^{\mathscr{H}}_{l,\kappa}$ are equivalent:

Theorem 12. *Let P and Q be two PEPA components and κ a function from PEPA components to $\mathbb{R}^+$. Then*

$$P \approx^{hc}_{l,\kappa} Q \text{ if and only if } P \approx^{\mathscr{H}}_{l,\kappa} Q.$$

Proof. First of all it should be proved that $P \approx^{hc}_{l,\kappa} Q$ implies $P \approx^{\mathscr{H}}_{l,\kappa}$. For this purpose, we prove that $\approx^{hc}_{l,\kappa}$ is a proportional bisimulation up to $\mathscr{H}$. This follows from the following cases.

- Let $\alpha \notin \mathscr{H} \cup \{\tau\}$. Since $P \approx^{hc}_{l,\kappa} Q$ it holds that for all $S \in \mathcal{C}/\approx^{hc}_{l,\kappa}$ and for all high context $C[\_]$, $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$. Since $\alpha \notin \mathscr{H} \cup \{\tau\}$, we have that $\frac{q[P,S,\alpha]}{\kappa_P} = \frac{q[Q,S,\alpha]}{\kappa_Q}$.

- Let $\alpha \in \mathscr{H} \cup \{\tau\}$. Since $P \approx^{hc}_{l,\kappa} Q$ it holds that for all $S \in \mathcal{C}/\approx^{hc}_{l,\kappa}$ such that $P, Q \notin S$ and for all high context $C[\_]$, $\frac{q_C[P,S,\tau]}{\kappa_P} = \frac{q_C[Q,S,\tau]}{\kappa_Q}$. If $C[\_]$ does not synchronize neither with P nor with Q, we have that $\frac{q[P,S,\tau]}{\kappa_P} = \frac{q[Q,S,\tau]}{\kappa_Q}$. On the other hand, let us consider a context $C[\_]$ with only one current action type $h \in \mathscr{H}$. Then, from $\frac{q_C[P,S,\tau]}{\kappa_P} = \frac{q_C[Q,S,\tau]}{\kappa_Q}$ and $\frac{q[P,S,\tau]}{\kappa_P} = \frac{q[Q,S,\tau]}{\kappa_Q}$, it follows that if P cooperates over h then also Q cooperates over h and $\frac{q[P,S,h]}{\kappa_P} = \frac{q[Q,S,h]}{\kappa_Q}$.

Now it is possible to show that if $P \approx^{\mathscr{H}}_{l,\kappa} Q$ then $P \approx^{hc}_{l,\kappa} Q$. In order to achieve this aim it is sufficient to prove that $\approx^{\mathscr{H}}_{l,\kappa}$ is a proportional bisimulation on high contexts. This follows from the following cases.

- Assume $\alpha \notin \mathscr{H} \cup \{\tau\}$. Since $P \approx^{\mathscr{H}}_{l,\kappa} Q$ it holds that for all $S \in \mathcal{C}/\approx^{\mathscr{H}}_{l,\kappa}$, $\frac{q[P,S,\alpha]}{\kappa_P} = \frac{q[Q,S,\alpha]}{\kappa_Q}$. Since a high context is only able to perform high level activities, we have that $q[P,S,\alpha] = q_C[P,S,\alpha]$ and $q[Q,S,\alpha] = q_C[Q,S,\alpha]$ for all high context $C[\_]$. Hence, $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$.

- Assume $\alpha = \tau$. It is possible to show that $\frac{q_C[P,S,\alpha]}{\kappa_P} = \frac{q_C[Q,S,\alpha]}{\kappa_Q}$ for all high level context $C[\_]$. We proceed by induction on the number of current action types of $C[\_]$ that synchronize with P and Q. Since $P \approx^{\mathscr{H}}_{l,\kappa} Q$ it holds that for all $S \in \mathcal{C}/\approx^{\mathscr{H}}_{l,\kappa}$ such that $P, Q \notin S$, $\frac{q[P,S,\tau]}{\kappa_P} = \frac{q[Q,S,\tau]}{\kappa_Q}$. If $C[\_]$ does not synchronize with P and Q we have that $q[P,S,\tau] = q_C[P,S,\tau]$ and $q[Q,S,\tau] = q_C[Q,S,\tau]$, i.e., $\frac{q_C[P,S,\tau]}{\kappa_P} = \frac{q_C[Q,S,\tau]}{\kappa_Q}$. Let $C[\_]$ be a context that has only one current action type $h \in \mathscr{H}$ that synchronizes with P and Q. From the fact that $P \approx^{\mathscr{H}}_{l,\kappa} Q$ it holds that for all $S \in \mathcal{C}/\approx^{\mathscr{H}}_{l,\kappa}$ such that $P, Q \notin S$, $\frac{q[P,S,h]}{\kappa_P} = \frac{q[Q,S,h]}{\kappa_Q}$ and $\frac{q[P,S,\tau]}{\kappa_P} = \frac{q[Q,S,\tau]}{\kappa_Q}$, we get $\frac{q_C[P,S,\tau]}{\kappa_P} = \frac{q_C[Q,S,\tau]}{\kappa_Q}$. The inductive step trivially follows.

□

Theorem 12 allows us to identify a local property of processes, with no quantification on the states and on the high contexts, which is a necessary and sufficient condition for *A-PSNI*. This is shown by the following corollary:

Corollary 12.1. *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$. Then*

$$P \in \textit{A-PSNI iff } P \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P.$$

Finally, we provide a characterization of *A-PSNI* in terms of *unwinding conditions*. In practice, whenever a state P' of a *A-PSNI* PEPA model P execute a high level activity leading it to a state P'', then P' and P'' are indistinguishable for a low level observer. Consider the following lines:

Theorem 13. *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$.*

$$P \in \textit{A-PSNI iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \textit{ implies } P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}.$$

Proof. First of all it is possible to prove that if $P \in$ *A-PSNI* then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}$. Indeed, by Proposition 14, $P' \in$ *A-PSNI* and therefore, by Corollary 11.1, $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P'$. By Definition 7.2.4 of $\approx^{\mathscr{H}}_{l,\kappa}$, for all $S \in \mathcal{C}/\approx^{\mathscr{H}}_{l,\kappa}$ such that $P' \setminus \mathscr{H}$, $P' \notin S$, both $\frac{q[P'\setminus\mathscr{H},S,\tau]}{\kappa_{P'\setminus\mathscr{H}}} = \frac{q[P',S,\tau]}{\kappa_{P'}}$ and $\frac{q[P'\setminus\mathscr{H},S,h]}{\kappa_{P'\setminus\mathscr{H}}} = \frac{q[P'S,h]}{\kappa_{P'}}$. Since $P' \setminus \mathscr{H}$ does not perform any high level action, $q[P' \setminus \mathscr{H}, S, h] = 0$ while, since $P' \xrightarrow{(h,r)} P''$,

$q[P', S, h] \neq 0$. Hence, from $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P'$, either h is not a current action type of P' or $P' \setminus \mathscr{H}$, $P' \in S$, i.e., $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P''$. Since also $P'' \in A\text{-}PSNI$, from $P'' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P''$ it follows that $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P'' \setminus \mathscr{H}$. Finally, since both $P' \setminus \mathscr{H}$ and $P'' \setminus \mathscr{H}$ do not perform any high level activity, $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P'' \setminus \mathscr{H}$ is equivalent to $P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}$.

Now it is possible to prove that if for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}$ then $P \in A\text{-}PSNI$. In particular, by Corollary 11.1 one should prove that $P \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P$. Let $\mathcal{R} = \{(P' \setminus \mathscr{H}, P'') \mid P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}\}$. If we show that $\mathcal{R}$ is a proportional bisimilarity up to $\mathscr{H}$, then we have the thesis. Indeed, one should observe that from $P' \setminus \mathscr{H} \approx^{\kappa}_{l} P'' \setminus \mathscr{H}$, for all $\alpha \notin \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ it is possible to get $\frac{q[P' \setminus \mathscr{H}, S, \alpha]}{\kappa_{P' \setminus \mathscr{H}}} = \frac{q[P'', S, \alpha]}{\kappa_{P''}}$. Finally, if $\alpha \in \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ we have $q[P' \setminus \mathscr{H}, S, \alpha] = 0$ and if $P'' \xrightarrow{(h,r)}$ then $P' \setminus \mathscr{H}$, $P'' \in S$. Therefore, $\mathcal{R}$ is a proportional bisimilarity up to $\mathscr{H}$. □

Using the relation $\approx^{\mathscr{H}}_{l,\kappa}$ property *A-PSNI* can also be characterized as follows.

Theorem 14. *Let P be a PEPA component and κ a function from PEPA components to $\mathbb{R}^+$.*

$$P \in A\text{-}PSNI \text{ iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \text{ implies } P' \approx^{\mathscr{H}}_{l,\kappa} P''.$$

Proof. First of all, it is possible to prove that if $P \in A\text{-}PSNI$ then for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \approx^{\mathscr{H}}_{l,\kappa} P''$. Indeed, by Proposition 14, $P' \in A\text{-}PSNI$ and hence, by Corollary 11.1, $P' \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P'$. By Definition 7.2.4 of $\approx^{\mathscr{H}}_{l,\kappa}$, for all $S \in \mathcal{C}/\approx^{\mathscr{H}}_{l,\kappa}$ such that $P' \setminus \mathscr{H}$, $P' \notin S$, both $\frac{q[P' \setminus \mathscr{H}, S, \tau]}{\kappa_{P' \setminus \mathscr{H}}} = \frac{q[P', S, \tau]}{\kappa_{P'}}$ and $\frac{q[P' \setminus \mathscr{H}, S, h]}{\kappa_{P' \setminus \mathscr{H}}} = \frac{q[P' S, h]}{\kappa_{P'}}$. Therefore, if $P' \setminus \mathscr{H}$, $P' \notin S$, since $P' \setminus \mathscr{H}$ is not able to perform any high level transition it holds $\frac{q[P' \setminus \mathscr{H}, S, h]}{\kappa_{P' \setminus \mathscr{H}}} = \frac{q[P' S, h]}{\kappa_{P'}} = 0$. Let S'' be the equivalence class of P''. Since $P' \xrightarrow{(h,r)} P''$ it holds $q[P', S'', h] \neq 0$ and it has to be $P' \in S''$.

Now it is possible to show that if for all $P' \in ds(P)$, $P' \xrightarrow{(h,r)} P''$ implies $P' \approx^{\mathscr{H}}_{l,\kappa} P''$, then $P \in A\text{-}PSNI$. In particular, by Corollary 11.1 there is the need to show that $P \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P$. Let $\mathcal{R} = \{(P' \setminus \mathscr{H}, P') \mid P' \in ds(P)\} \cup \approx^{\mathscr{H}}_{l,\kappa}$.

If we prove that $\mathcal{R}$ is a proportional bisimilarity up to $\mathscr{H}$, then we have the thesis. Indeed, one should observe that for all $\alpha \notin \mathscr{H}$ and for all P'' it holds $q(P' \setminus \mathscr{H}, P'' \setminus \mathscr{H}, \alpha) = q(P', P'', \alpha)$, so for all $S \in \mathcal{C}/\mathcal{R}$ we get $q[P' \setminus \mathscr{H}, S, \alpha] = q[P', S, \alpha]$. In the end, if $\alpha \in \mathscr{H}$, for all $S \in \mathcal{C}/\mathcal{R}$ we have $q[P' \setminus \mathscr{H}, S, \alpha] = 0$. Since $\approx_l^{\mathscr{H}} \subseteq \mathcal{R}$ by the hypothesis we have that if $P' \notin S$, then $q[P', S, \alpha] = 0$. Hence, $\mathcal{R}$ is a proportional bisimilarity up to $\mathscr{H}$. □

Corollary 12.1 and Theorems 13 and 14 provide different characterizations of *A-PSNI* which naturally lead to efficient methods for the verification and construction of secure systems.

Example 1. Consider again the example in which a web server is serving the requests coming from a client, as presented for the *PSNI* case. Exactly as before, the client first makes a request through the activity *req* with rate ρ. The server can directly reply with type *res* and rate μ or a it can redirect to some other server, which processes a high level authentication with type *log* and rate λ; then a reply is sent to the client with type *res*, but this time with rate γ, which is $\gamma \neq \mu$. The rates assume the following quantities: $\mu = 0.4$ and $\gamma = 2\mu$. Moreover, we will consider the following proportional factors: κ $\kappa_{P_2 \setminus \mathscr{H}} = 2$, $\kappa_{P_3 \setminus \mathscr{H}} = 4$. In this example, we will show that this system satisfies *A-PSNI*. Formally, the system has the following *PEPA* specification:

$$P_1 \stackrel{\text{def}}{=} (res, \rho).P_2$$
$$P_2 \stackrel{\text{def}}{=} (res, \mu).P_1$$
$$P_2 \stackrel{\text{def}}{=} (log, \lambda).P_3$$
$$P_3 \stackrel{\text{def}}{=} (res, \gamma).P_1$$

and its derivation graph is depicted in Figure 7.1 here below.

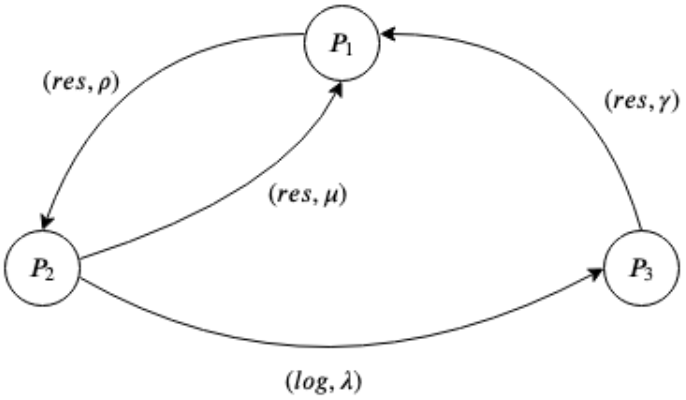

Figure 7.1: A simple three state model

By following Theorem 13, we can prove that $P_1 \in A\text{-}PSNI$. Indeed, it is easy to prove that $P_2 \setminus \mathscr{H} \approx_l^\kappa P_3 \setminus \mathscr{H}$ when $\approx_l^\kappa$ is the proportional bisimilarity, with κ a function from PEPA components to $\mathbb{R}^+$. Indeed, in order to prove that $P_1 \in PSNI$ we must check if $P_2 \setminus \mathscr{H} \approx_l^\kappa P_3 \setminus \mathscr{H}$, which consists in the following equation: for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\approx_l^\kappa$,

$$\frac{q[P_2 \setminus \mathscr{H}, S, \alpha]}{\kappa_{P_2 \setminus \mathscr{H}}} = \frac{q[P_3 \setminus \mathscr{H}, S, \alpha]}{\kappa_{P_3 \setminus \mathscr{H}}}$$

The only activity to be considered is *res*, so the computation will continue as follows:

$$\frac{\mu}{\kappa_{P_2 \setminus \mathscr{H}}} = \frac{\gamma}{\kappa_{P_3 \setminus \mathscr{H}}}$$

and this is true for the quantities given in this example.

Now suppose that P_2 always synchronizes on *log*. Then for a low level observer, the system behaves as $P_1/\mathscr{H}$ as depicted in Figure 7.2 here below.

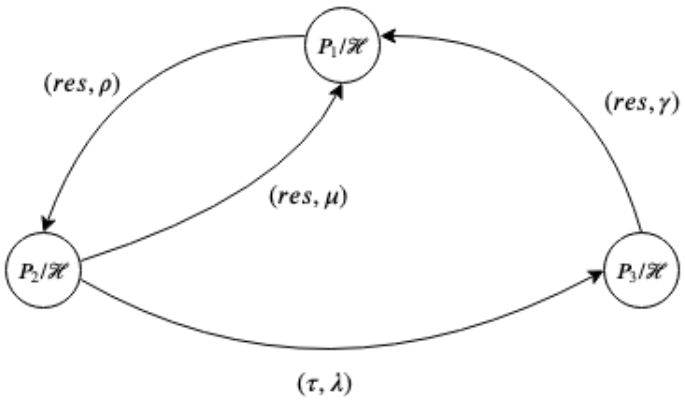

Figure 7.2: The model of $P_1/\mathscr{H}$

We can compute the steady-state distribution of $P_1/\mathscr{H}$ by solving the global balance equations together with the normalization condition, obtaining:

$$\begin{aligned} \pi\rho &= \pi_2\mu + \pi_3\gamma \\ \pi_2(\lambda+\mu) &= \pi_1\rho \\ \pi_3\gamma &= \pi_2\lambda \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

whose solution is

$$\begin{aligned} \pi_1 &= \frac{\mu\gamma + \lambda\gamma}{\mu\gamma + \lambda\gamma + \lambda\rho} \\ \pi_2 &= \frac{\lambda\rho\gamma}{(\lambda+\gamma)(\mu\gamma + \lambda\gamma + \lambda\rho)} \\ \pi_3 &= \frac{\lambda^2\rho}{(\lambda+\gamma)(\mu\gamma + \lambda\gamma + \lambda\rho)} \end{aligned}$$

where π_1, π_2 and π_3 denote the steady-state probabilities of states $P_1/\mathscr{H}$, $P_2/\mathscr{H}$ and $P_3/\mathscr{H}$, respectively.

Consider now the case in which P_2 never synchronizes over *log*. Then, the low level view of the system is represented by $P_1 \setminus \mathscr{H}$ depicted in Figure 7.3 here below.

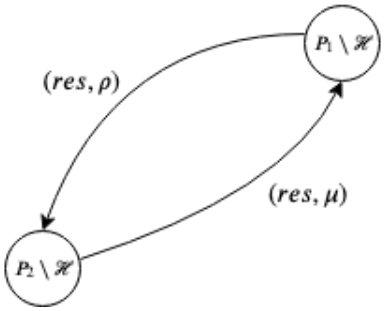

Figure 7.3: The model of $P_1 \setminus \mathscr{H}$

Again, we can compute the steady-state distribution of $P_1 \setminus \mathscr{H}$ by solving the global balance equations with the normalization condition, and the results are the same of those obtained for *PSNI* case.

Chapter 8

A Decision Algorithm for Approximate-Persistent Stochastic Non-Interference

8.1 Overview

So far, we have been studying the notions of *PSNI* and consequently its variant *A-PSNI* based on proportional bisimulation, by analysing their different characterizations in order to avoid the use of universal quantifications. It might be really interesting and useful to have not only the theoretical knowledges about *A-PSNI* but also some practical algorithm which allows one to effectively decide whether a PEPA component satisfies *Approximate-Persistent Stochastic Non-Interference* property. The authors of [37, 36] have already introduced an algorithm for *PSNI* case, and therefore, in this section we want to reformulate it in order to provide a decision algorithm for *A-PSNI*.
Recall that a decision problem is a problem having only to possible outputs, usually *yes* or *no*. In other words, a decision problem can always be posed as a yes-no question of the input values. Hence, the algorithm which will be presented in this section will answer to the following question: "*is the PEPA component P satisfying the property PSNI?*" and the answer will naturally be *yes* or *no*. More specifically, as the algorithm introduced in the Section 6 of [37], the decision algorithm for *A-PSNI* presented in this section will take in

input two PEPA components P and Q, each associated with finite derivation graphs, and it will decide if $P \approx^{\mathscr{H}}_{l,\kappa} Q$, and by exploiting Corollary 12.1 we are allowed to decide whether a process is *A-PSNI*.

The proposed algorithm is founded on the label-compatibility problem, in particular the decision problem of deciding whether $P \approx^{\mathscr{H}}_{l,\kappa} Q$ is mapped into a label-compatibility problem. Since a label-compatibility problem takes in input a directed labelled weighted graph, first of all we should define it. Consider the following definition:

Definition 8.1.1. (Directed labelled weighted graph) *A directed labelled weighted graph is a tuple $G = (V, Lab, E, w)$ where:*

- *V is a finite set of vertices;*
- *Lab is a finite set of labels;*
- *$E \subseteq V \times V \times Lab$ is a finite set of labelled edges;*
- *$w : E \to \mathbb{R}$ is a weighting function that associates a value to each edge.*

Given $V' \subseteq V$, with $w(v, V', a)$ the sum of the weights of the edges from v to V' having label a will be indicated.

Now consider the following definition of the label-compatibility problem introduced in [5] which extends the one introduced by [55] to directed labelled weighted graph.

Definition 8.1.2. (Label-Compatibility Problem) *Let $G = (V, Lab, E, w)$ be a directed labelled weighted graph and $\mathcal{R} \subseteq V \times V$ be an equivalence relation over V. $\mathcal{R}$ is said to be* label-compatible *with G if for each $a \in Lab$, for each $C, C' \in V/\mathcal{R}$, and for each $v, v' \in C$ it holds that $w(v, C', a) = w(v', C', a)$.*
Let $G = (V, Lab, E, w)$ be a directed labelled weighted graph: The labelled weighted compatibility problem over G *requires to compute the largest equivalence relation label-compatible with G.*

Notice that in [5] the authors proved that the label-compatibility problem always has a unique solution.

Now, the remaining thing is the definition of the input to give to the algorithm. The label-compatibility problem defined here above takes in input a

directed labelled weighted graph and we want that the latter problems solves the decision problem of whether $P \approx^{\mathscr{H}}_{l,\kappa} Q$. Hence, we need to introduce a new suitable graph. Consider the next definition:

Definition 8.1.3. (Up to $\mathscr{H}$ Proportional Lumping Graph) *Let P and Q be PEPA components and let κ be a function from PEPA components to $\mathbb{R}^+$. The* up to $\mathscr{H}$ proportional lumping graph *of $P \cup Q$ is the directed labelled weighted graph $\mathscr{L}\mathscr{H}_{P\cup Q} = (V_{P\cup Q}, E_{P\cup Q}, w_{P\cup Q})$, where:*

- *$V_{P\cup Q}$ is $ds(P) \cup ds(Q)$*
- *$E_{P\cup Q}$ is the set of labelled edges*

$$E_{P\cup Q} = \{(R, R', \alpha) \mid R \xrightarrow{(\alpha,r)} R'\} \cup \{(R, R, \alpha) \mid \text{and } \alpha \in \mathscr{H} \cup \{\tau\}\}$$

  with R and R' in $V_{P\cup Q}$

- *$w_{P\cup Q}$ is the function which associates to each edges in $E_{P\cup Q}$ the value*

$$w_{P\cup Q}(R, R', \alpha) = \begin{cases} q(R, R', \alpha)/\kappa_R & \text{if } \alpha \notin \mathscr{H} \cup \{\tau\} \vee R \neq R' \\ -q[R, V_{P\cup Q} \setminus \{R\}, \alpha]/\kappa_R & \text{otherwise} \end{cases}$$

When P and Q coincide we use $\mathscr{L}\mathscr{H}_P$ to denote $\mathscr{L}\mathscr{H}_{P\cup P}$.

Then, consider the next theorem, which will allow one to exploit a variant of the algorithm presented by the authors of [5].

Theorem 15. *Let P and Q be two PEPA components and let κ be a function from PEPA components to $\mathbb{R}^+$. It holds that $P \approx^{\mathscr{H}}_{l,\kappa} Q$ if and only if in the largest equivalence relation label-compatible with $\mathscr{L}\mathscr{H}_{P\cup Q}$ the vertices P and Q are equivalent.*

Proof. Consider $\mathcal{R}$, the largest relation label-compatible with $\mathscr{L}\mathscr{H}_{P\cup Q}$. We are going to show that $\mathcal{R} = \approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q))$ by proving that $\mathcal{R} \subseteq \approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q))$ and $\approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q)) \subseteq \mathcal{R}$.
For the purpose of proving that $\mathcal{R} \subseteq \approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q))$ it is sufficient to prove that $\mathcal{R}$ is a proportional bisimulation up to $\mathscr{H}$. Assume $(R_1, R_2) \in \mathcal{R}$ and $C \in V_{P\cup Q}/\mathcal{R}$. If $\alpha \notin \mathscr{H} \cup \{\tau\}$, then $\sum_{R'\in C} q(R_1, R', \alpha)/\kappa_{R_1} = q[R_1, C, \alpha]/\kappa_{R_1} = w_{P\cup Q}(R_1, C, \alpha) = w_{P\cup Q}(R_2, C, \alpha) = q[R_2, C, \alpha]/\kappa_{R_2} = \sum_{R'\in C} q(R_2, R', \alpha)/\kappa_{R_2}$. Similarly, if $\alpha \in \mathscr{H} \cup \{\tau\}$ and $R_1, R_2 \notin C$ we get the

thesis, since $R_i \notin C$ implies $q[R_i, C, \alpha]/\kappa_{R_i} = w_{P\cup Q}(R_i, C, \alpha)$, for $i = 1, 2$. For the purpose of proving that $\approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q)) \subseteq \mathcal{R}$ it is sufficient to show that $\approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q)) \subseteq \mathcal{R}$ is label-compatible. Assume $R_1 \approx^{\mathscr{H}}_{l,\kappa} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q))R_2$, and $C \in V_{P\cup Q}/\approx^{\mathscr{H}}_{l} \cap(\mathcal{D}(P) \cup \mathcal{D}(Q))$, we have to prove that $w_{P\cup Q}(R_1, C, \alpha) = w_{P\cup Q}(R_2, C, \alpha)$, for each $\alpha \in \mathcal{A}$. In case $\alpha \in \mathscr{H} \cup \{\tau\}$, $w_{P\cup Q}(R_1, C, \alpha) = \sum_{R'\in C, R'\neq R_1} q(R_1, R', \alpha)/\kappa_{R_1} - q[R_1, V_{P\cup Q} \setminus \{R_1\}, \alpha]/\kappa_{R_1} = -\sum_{R'\notin C} q(R_1, R', \alpha)/\kappa_{R_1} = -\sum_{C'\neq C} w_{P\cup Q}(R_1, C', \alpha)/\kappa_{R_1}$. Since, neither R_1 nor R_2 belongs to any of the classes C' involved in the last sum, we get $-\sum_{C'\neq C} w_{P\cup Q}(R_1, C', \alpha) = -\sum_{C'\neq C} w_{P\cup Q}(R_2, C', \alpha) = w_{P\cup Q}(R_2, C, \alpha)$. □

Now, it is possible to present the two algorithms in order to decide if $P \approx^{\mathscr{H}}_{l,\kappa} Q$: one is the algorithm for Labelled Weighted Compatibility, the other is the algorithm for $\approx^{\mathscr{H}}_{l,\kappa}$, proportional bisimulation up to $\mathscr{H}$. The algorithm for Labelled Weighted Compatibility consists in a function named $LWD_T(-)$ which takes in input a directed labelled weighted graph. $LWD_T(-)$ will produce as output the partition of the input graph nodes according to the largest equivalence relation label-compatible with the input graph. The algorithm $LWD_T(-)$ is exploited within the other algorithm for $\approx^{\mathscr{H}}_{l,\kappa}$, which consists in a function named $LBup\mathscr{H}(-,-)$ which decides whether $P \approx^{\mathscr{H}}_{l,\kappa} Q$.

8.2 Algorithms and Complexity

In this section, pseudo-code for algorithms $LWD_T(-)$ and $LBup\mathscr{H}(-,-)$ are presented and then analysis on their computational complexity are reported. Notice that the algorithm is the same as the one presented in [37].

Consider the next corollary which shows that the algorithm $LBup\mathscr{H}(-,-)$ is able to solve the considered problem in time $O(mlog(n))$ where n is the number of states and m the number of transitions in the Markov chain, as in the article [55].

Corollary 15.1. *Let P and Q be PEPA components and let κ be a function from PEPA components to $\mathbb{R}^+$. Let $\mathscr{L}\mathscr{H}_{P\cup Q} = (V_{P\cup Q}, \mathcal{A}, E_{P\cup Q}, w_{P\cup Q})$ be the up to $\mathscr{H}$ lumping graph of $P \cup Q$. $LBup\mathscr{H}(P, Q)$ decides $P \approx^{\mathscr{H}}_{l,\kappa}$ in time $O(|V_{P\cup Q}|+|E_{P\cup Q}|log|V_{P\cup Q}|)$.*

According to Corollary 12.1, in order to prove that a component P is *PSNI*

we should verify that $P \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P$. Therefore, by combining Corollary 12.1 and Corollary 15.1 we are allowed to solve the decision problem by calling $LBup\mathscr{H}(P \setminus \mathscr{H}, P)$. This clearly requires the creations of graphs $\mathcal{D}(P \setminus \mathscr{H})$ and $\mathcal{D}(P)$. Nevertheless, the following theorem shows that the decision algorithm for *A-PSNI* can work only with $\mathcal{D}(P)$, by exploiting results obtained by Theorem 14 and Theorem 15:

Theorem 16. *Let P be PEPA components. Let $Comp_P$ be the largest equivalence relation label-compatible with $\mathscr{L}\mathscr{H}_P$. P is* PSNI *if and only if whenever $P' \xrightarrow{(h,r)} P''$ with $P' \in ds(P)$ and $h \in \mathscr{H}$ it holds that $(P', P'') \in Comp_P$.*

Proof. According to Theorem 13, P is *PSNI* if and only if for each $P' \in ds(P)$ if $P' \xrightarrow{(h,r)} P''$ with $h \in \mathscr{H}$, then $P' \approx^{\mathscr{H}}_{l,\kappa} P''$. By Theorem 15 this holds if and only if for each $P' \in ds(P)$ if $P' \xrightarrow{(h,r)} P''$ with $h \in \mathscr{H}$, then in the largest equivalence relation label-compatible with $\mathscr{L}\mathscr{H}_P$ the vertices P' and P'' are equivalent.

Thanks to the latter result, the computation of $\mathcal{D}(P \setminus \mathscr{H})$ can be avoided, therefore the computational complexity is greatly reduced. See the following pages for the specification of the algorithms $LWD_T(-)$ and $LBup\mathscr{H}(-,-)$.

Algorithm 1 Algorithm for Labelled Weighted Compatibility (Part 1)

```
1: function LCW_T(G = (V, Lab, E, w))
2:     P = {V}
3:     UB = P
4:     TB = ∅
5:     w[v] = unused for every v ∈ V
6:     while UB ≠ ∅ do
7:         C = Pop(UB)
8:         for l ∈ Lab, v' ∈ C do
9:             pre[v', l] = store the pre-image of v' with respect to l-edges
10:        end for
11:        for l ∈ Lab do
12:            TS = ∅
13:            for v' ∈ C, v ∈ pre[v', l] do
14:                if w[v] == unused then
15:                    TS = TS ∪ {v}
16:                    w[v] = w(v, v', l)
17:                else
18:                    w[v] = w[v] + w(v, v', l)
19:                end if
20:                for v ∈ TS do
21:                    if w[v] ≠ 0 then
22:                        B = GetBlockOf(v)
23:                        if B contains 0 marked states then
24:                            TB = TB ∪ {B}
25:                        end if
26:                        mark v in B
27:                    end if
28:                end for
29:                while TB ≠ ∅ do
30:                    B = Pop(TB)
31:                    B_1 = marked states in B
32:                    B = remaining states in B
33:                    if B == ∅ then
34:                        give identity of B to B_1 in P
35:                    else
36:                        make B_1 a new block in P
37:                    end if
```

Algorithm 2 Algorithm for Labelled Weighted Compatibility (Part 2)

38: $y = PMC(w[v])$ *for* $v \in B_1$
39: $B_2 = \{v \in B_1 \mid w[v] \neq y\}$
40: $B_1 = B_1 \setminus B_2$
41: **if** $B_2 == \emptyset$ **then**
42: $m = 1$
43: **else**
44: *sort and partition* B_2 *according to* $w[v]$
45: *make each of* $B_2, ..., B_m$ *a new block in* $\mathcal{P}$
46: **end if**
47: **if** $B \in UB$ **then**
48: *add* $B_1, ..., B_m$ *except* B *in* UB
49: **else**
50: *add* $[B,]^? B_1, ..., B_m$ *except largest in* UB
51: **end if**
52: **end while**
53: **for** $v \in TS$ **do**
54: $w[v] = unused$
55: **end for**
56: **end for**
57: **end for**
58: **end while**
59: **return** $\mathcal{P}$
60: **end function**

Algorithm 3 Algorithm for $\approx^{\mathscr{H}}_{l,\kappa}$

1: **function** $LBup\mathscr{H}(P, Q)$
2: Compute $\mathscr{L}\mathscr{H}_{P\cup Q}$
3: $Comp_{P\cup Q} = LCW\_T(\mathscr{L}\mathscr{H}_{P\cup Q})$
4: **return** $(P, Q) \in Comp_{P\cup Q}$
5: **end function**

Chapter 9

Conclusion

In this final chapter, we would like to summarize the contributions given within this thesis by further analysing their possible impacts, and discuss some idea of topics for future works.

9.1 Summary

The central issue which has been addressed in this thesis is the problem of introducing a variant of *Persistent Stochastic Non-Interference* (*PSNI*) security property for stochastic, cooperating processes expressed as terms of the *Performance Evaluation Process Algebra* (PEPA), in such a way that it can be widely adopted in several suitable real world situations. Indeed, *PSNI* results to be a very useful property in order to ensure the security of a system, but it is based upon a bisimulation-like relation called lumpable bisimilarity, and the latter equivalence relation considers two PEPA components lumpably bisimilar if the transition rates from these components to any equivalence class are the same. This kind of requirement is quite difficult to be satisfied if we try to model real world systems, since the transition rates between two states are hardly exactly the same. However, we have seen that the necessary and sufficient condition for proving that a component P is *PSNI*, we should verify $P \setminus \mathscr{H} \approx_l^{\mathscr{H}} P$, where $\mathscr{H}$ is the set of high level action types.

In order to cope with this problem, the idea proposed in this thesis is the relaxation of the *PSNI* property by exploiting another bisimulation-like relation different from lumpable bisimulation but the one which approximates the

latter equivalence relation. In this thesis, after reviewing some fundamental concepts regarding to Continuous Time Markov Chain theory including the definition of strong lumpability and its variant lumpable bisimulation, new notions of lumpability have been proposed, the quasi-lumpability and proportional lumpability. Both of the concepts deal with the Markov chain perturbation theory, in which small differences on the transition rates of the considered Markov chain are allowed. Quasi-lumpability requires not the equation of transition rates from components to equivalence classes, but it only asks the absolute value of differences between total conditional transition rates to be less than some bound ϵ. In this way, even if the rates are not exactly the same, we can still obtain a lumping of the Markov chain in approximate way.
Conversely, proportional lumpability requires the equation of the transition rates from components to equivalence classes, divided by some real number κ which depends on components. Then, in Chapter 4 and Chapter 5 several equivalence relations over PEPA components have been proposed. Strong equivalence and its lumpable bisimulation are both based on strong lumpability, and in particular lumpable bisimulation is the one involved in the definition of the property *PSNI*. Subsequently, their approximate version have been proposed, namely, quasi-lumpable bisimulation and proportional bisimulation which are based on quasi-lumpability and proportional lumpability respectively. Nevertheless, quasi-lumpable bisimulation resulted not to be the most suitable equivalence relation in order to define a *PSNI*-kind property, since it is not preserved under union, in the sense that the union of two quasi-lumpable bisimulations is still a quasi-lumpable bisimulation but not with respect to the same bound ϵ.
Proportional bisimulation does not suffer of this kind of problem, so that it has been chosen as the equivalence relation for the new *PSNI* property, which has been named *Approximate-PSNI*, shortened as *A-PSNI*. In Chapter 7, *A-PSNI* has been studied in its details by following exactly the same steps as *PSNI* case shown in the article [37] and in particular two characterizations of *A-PSNI* have been defined: the first involves a single bisimulation-like equivalence check, while the second is formulated in terms of unwinding conditions. In Chapter 8, a decision algorithm for *A-PSNI* has been proposed, as like as in the article [36]. The algorithm focuses on showing if P is *PSNI*, by proving that $P \setminus \mathscr{H} \approx^{\mathscr{H}}_{l,\kappa} P$. The algorithm is based on labelled weighted compatibility problem, which consists in a variant of the one presented in [55].

As we have seen in Chapter 6, *PSNI* consists in a quantitative extension of

the *Non-Interference*, which is an information flow security property that controls unwanted information flow to undesired external users, by assuming that the external observers can measure the timing behaviour of the system. In this context, the process algebra PEPA resulted to be the most suitable language in order to model these kind of systems because it allows to specify delays so that the quantitative properties of the considered system can be easily modelled. Notice that the definition of *PSNI* proposed in article [37] relies on the assumption that the external observer can recognize *any* execution path with its delays, meaning that the whole transient behaviour of the system is entirely seen by the observer. This consists in a very strict situation, which is quite difficult to obtain in real world situations, fortunately. Therefore, by following this reasoning we can say that a system which satisfies *PSNI* is rare but we are ensured that it is surely secure. Since the original *PSNI* is a strong property, we can think that one is allowed to lighten the assumptions and conditions, and this has originated our *A-PSNI*, so that even some approximations on rates are introduced, the system can be considered safe enough. The introduction of *A-PSNI* allows one to apply it to a wider set of modelled systems, also to those which have been considered not secure enough by *PSNI* but still they are acceptably safe.

9.2 Directions for future works

In [45] a partitioning strategy for PEPA components which involves the use of a spectral clustering algorithm that minimizes an upper bound for approximate strong equivalence has been proposed. In particular, the set of components to be partitioned is associated to a weighted undirected graph and then according to it a pairwise similarity matrix can be constructed. By exploiting the *Laplacian* of the similarity matrix, the algorithm is going to select the n eigenvectors that correspond to the n largest eigenvalues of the considered Laplacian matrix. The authors of [45] then have experimentally applied and evaluated over a realistic case study, inspired by an issue raised for the *Heroku PaaS* (*Platform as a Service*) provider by showing how the accuracy of reduced model is approaching the results obtained by the original model. As like as this case, as an extension of this thesis, also a partitioning strategy for PEPA components based on the use of proportional bisimulation can be proposed, in order to experimentally check what we have theoretically analysed until now and compare the results on performances of the original non-reduced system

with respect to the reduced system by proportional bisimulation equivalence relation strategy.

Ringraziamenti

Ringrazio la Prof.ssa Sabina Rossi per avermi seguita pazientemente durante gli ultimi sei mesi, aiutandomi attivamente nella realizzazione della tesi in modo da riuscire a consegnarla a giugno.

Ringrazio i miei genitori che mi supportano sempre e mi hanno sopportato in questi ultimi mesi durante la stesura della tesi, caratterizzati da continuo alternarsi di situazioni positive e negative.

Ringrazio i miei amici, soprattutto gli amici membri del gruppo *KAFF-e* che mi hanno sempre aiutato in questi cinque anni della mia carriera universitaria. Senza di voi non sarei mai arrivata qui.

Ringrazio Margherita, l'amica con cui ho condiviso la fatica, lo sforzo, le seccature, i progetti che non funzionano, *ecc.* Il supporto morale a vicenda è stato fondamentale durante questi due anni della magistrale.

Bibliography

[1] Martín Abadi, Bruno Blanchet, and Cédric Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *J. ACM*, 65(1):1:1–1:41, 2018.

[2] Ivo Adan and Jacques Resing. *Queueing Theory*. Eindhoven University of Technology. Department of Mathematics and Computing Science, 2001.

[3] Alessandro Aldini and Marco Bernardo. A general framework for nondeterministic, probabilistic, and stochastic noninterference. In *Foundations and Applications of Security Analysis, Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security, ARSPA-WITS 2009, York, UK, March 28-29, 2009, Revised Selected Papers*, pages 18–33, 2009.

[4] Michael Alexander and William Gardner. *Process Algebra for Parallel and Distributed Processing*. Chapman & Hall/CRC, 2008.

[5] Giacomo Alzetta, Andrea Marin, Carla Piazza, and Sabina Rossi. Lumping-based equivalences in markovian automata: Algorithms and applications to product-form analyses. *Inf. Comput.*, 260:99–125, 2018.

[6] Jos C. M. Baeten and Mario Bravetti. A generic process algebra. *Electr. Notes Theor. Comput. Sci.*, 162:65–71, 2006.

[7] Jan A. Bergstra. *Handbook of Process Algebra*. Elsevier Science Inc., 2001.

[8] Marco Bernardo, Lorenzo Donatiello, and Roberto Gorrieri. Modeling and analyzing concurrent systems with mpa. In *Proc. of 2nd Process Algebra and Performance Modelling Workshop*, pages 175–189, 1994.

[9] Andrew Bortz and Dan Boneh. Exposing private information by timing web applications. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 621–628, 2007.

[10] Annalisa Bossi, Riccardo Focardi, Carla Piazza, and Sabina Rossi. A proof system for information flow security. In *Logic Based Program Synthesis and Tranformation, 12th International Workshop, LOPSTR 2002, Madrid, Spain, September 17-20,2002, Revised Selected Papers*, pages 199–218, 2002.

[11] Annalisa Bossi, Carla Piazza, and Sabina Rossi. Compositional information flow security for concurrent programs. *Journal of Computer Security*, 15(3):373–416, 2007.

[12] David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

[13] Peter Buchholz. Exact and ordinary lumpability in finite markov chains. *Journal of Applied Probability*, 31, 1995.

[14] Michele Bugliesi and Sabina Rossi. Non-interference proof techniques for the analysis of cryptographic protocols. *Journal of Computer Security*, 13(1):87–113, 2005.

[15] Grace E. Cho and Carl D. Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1):137–150, 2001.

[16] Bong Wan Choi, Way Kuoj, and K. John K. Jackman. Petri net extensions for modelling and validating manufacturing systems. *The International Journal Of Production Research*, 32:1819–1835, 1994.

[17] Silvia Crafa and Sabina Rossi. P-congruences as non-interference for the pi-calculus. In *Proceedings of the 2006 ACM workshop on Formal methods in security engineering, FMSE 2006, Alexandria, VA, USA, November 3, 2006*, pages 13–22, 2006.

[18] Silvia Crafa and Sabina Rossi. Controlling information release in the pi-calculus. *Inf. Comput.*, 205(8):1235–1273, 2007.

[19] Rocco De Nicola. Behavioral equivalences. In *Encyclopedia of Parallel Computing*, pages 120–127. 2011.

[20] Susanna Donatelli, Marina Ribaudo, and Jane Hillston. A comparison of performance evaluation process algebra and generalized stochastic petri nets. In *Proceedings of the Sixth International Workshop on Petri Nets and Performance Models, PNPM 1995, Durham, NC, USA, October 3-6, 1995*, pages 158–168, 1995.

[21] Edward W. Felten and Michael A. Schneider. Timing attacks on web privacy. In *CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 1-4, 2000.*, pages 25–32, 2000.

[22] Riccardo Focardi and Roberto Gorrieri. A classification of security properties for process algebras. *Journal of Computer Security*, 3(1):5–33, 1994/1995.

[23] Riccardo Focardi and Roberto Gorrieri. Classification of security properties (part I: information flow). In *Foundations of Security Analysis and Design, Tutorial Lectures [revised versions of lectures given during the IFIP WG 1.7 International School on Foundations of Security Analysis and Design, FOSAD 2000, Bertinoro, Italy, September 2000]*, pages 331–396, 2000.

[24] Riccardo Focardi, Roberto Gorrieri, and Fabio Martinelli. Non interference for the analysis of cryptographic protocols. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 354–372, 2000.

[25] Riccardo Focardi, Roberto Gorrieri, and Fabio Martinelli. Real-time information flow analysis. *IEEE Journal on Selected Areas in Communications*, 21(1):20–35, 2003.

[26] Riccardo Focardi, Sabina Rossi, and Andrei Sabelfeld. Bridging language-based and process calculi security. In *Foundations of Software Science and Computational Structures, 8th International Conference, FOSSACS 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, pages 299–315, 2005.

[27] Wan Fokkink. *Introduction to Process Algebra.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2000.

[28] Giuliana Franceschinis and Richard R. Muntz. Bounds for quasi-lumpable markow chains. *Perform. Eval.*, 20(1-3):223–243, 1994.

[29] Robert E. Funderlic and Carl D. Meyer Jr. Sensitivity of the stationary distribution vector for an ergodic markov chain. *Linear Algebra and its Applications*, 76:1–17, 1986.

[30] Joseph A. Goguen and José Meseguer. Security policies and security models. pages 11–20. IEEE Computer Society Press, 1982.

[31] Roberto Gorrieri, Enrico Locatelli, and Fabio Martinelli. A simple language for real-time cryptographic protocol analysis. In *Programming Languages and Systems, 12th European Symposium on Programming, ESOP 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, pages 114–128, 2003.

[32] Norbert Götz, Ulrich Herzog, and Michael Rettelbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras. In *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93, Santa Clara, CA, USA, May 10-14, 1993*, pages 121–146, 1993.

[33] Moshe Haviv and Ludo Van Der Heyden. Perturbation bounds for the stationary probabilities of a finite markov chain. *Advances in Applied Probability*, 16(4):804–818, 1984.

[34] Matthew Hennessy and James Riely. Information flow vs. resource access in the asynchronous pi-calculus. *ACM Trans. Program. Lang. Syst.*, 24(5):566–591, 2002.

[35] Jane Hillston. *A compositional approach to performance modelling.* Cambridge Press, 1996.

[36] Jane Hillston, Andrea Marin, Carla Piazza, and Sabina Rossi. Information flow security for stochastic processes. In *Computer Performance Engineering - 15th European Workshop, EPEW*, pages 142–156, 2018.

[37] Jane Hillston, Andrea Marin, Carla Piazza, and Sabina Rossi. Persistent stochastic non-interference. *Information and Computation*, 2019. To appear.

[38] Jane Hillston, Andrea Marin, Sabina Rossi, and Carla Piazza. Contextual lumpability. In *7th International Conference on Performance Evaluation Methodologies and Tools, ValueTools '13, Torino, Italy, December 10-12, 2013*, pages 194–203, 2013.

[39] Ilse Ipsen and Carl D. Meyer Jr. Uniform stability of markov chains. volume 15, 1995.

[40] Carl D. Meyer Jr. The condition of a finite markov chain and perturbation bounds for the limiting probabilities. *SIAM J. Matrix Analysis Applications*, 1(3):273–283, 1980.

[41] John G. Kemeny and James Laurie Snell. *Finite markov chains*. Springer New York, 1960.

[42] Heiko Mantel. Unwinding possibilistic security properties. In *Computer Security - ESORICS 2000, 6th European Symposium on Research in Computer Security, Toulouse, France, October 4-6, 2000, Proceedings*, pages 238–254, 2000.

[43] Andrea Marin and Sabina Rossi. On the relations between markov chain lumpability and reversibility. *Acta Inf.*, 54(5):447–485, 2017.

[44] John McLean. A general theory of composition for trace sets closed under selective interleaving functions. In *1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, May 16-18, 1994*, pages 79–93, 1994.

[45] Dimitrios Milios and Stephen Gilmore. Component aggregation for PEPA models: An approach based on approximate strong equivalence. *Perform. Eval.*, 94:43–71, 2015.

[46] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[47] Peter Y. A. Ryan and Steve A. Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1/2):75–103, 2001.

[48] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.

[49] Eugene Seneta. Perturbation of the stationary distribution measured by ergodicity coefficients. *Advances in Applied Probability*, 20(1):228–230, 1988.

[50] Eugene Seneta. Sensitivity analysis, ergodicity coefficients and rank-one updates for finite markov chains. *Numerical Solution of Markov Chains, W.J. Stewart (ed.), Marcel Dekker*, (1):121–129, 1991.

[51] Geoffrey Smith and Dennis M. Volpano. Secure information flow in a multi-threaded imperative language. In *POPL '98, Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Diego, CA, USA, January 19-21, 1998*, pages 355–364, 1998.

[52] Eilon Solan and Nicolas Vieille. Perturbed markov chains. *Journal of Applied Probability*, 40(1):107–122, 2003.

[53] William J. Stewart. *Introduction to the numerical solution of Markov Chains*. Princeton University Press, 1994.

[54] Ushio Sumita and Maria Rieders. Lumpability and time reversibility in the aggregation-disaggregation method for large markov chains. *Communications in Statistics. Stochastic Models*, 5(1):63–81, 1989.

[55] Antti Valmari and Giuliana Franceschinis. Simple $O(m \log n)$ time markov chain lumping. 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*, pages 38–52, 2010.

[56] J. Todd Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. In *Proceedings of the 1990 IEEE Symposium on Security and Privacy, Oakland, California, USA, May 7-9, 1990*, pages 144–161, 1990.