Chapitre 8

Quantitative verification of Markov chains

8.1. Introduction

Hardware and software systems are more and more pervasive in every day life and, therefore, there is an obvious demand for these systems to meet the functional and performance requirements that the users expect. Automatic verification methods are a possible, and doable, way to increase our level of confidence in the systems that we design and produce, both in terms of functionality (what the system does) and performance (how long does it take). Verification methods that take into account the randomness of systems work with a model of the system which is a stochastic process. In order to limit the complexity of the verification process, these stochastic processes are often either Discrete Time Markov Chains (DTMC) or Continous Time Markov Chains (CTMC), usually automatically generated by some higher level formalism like stochastic Petri nets or stochastic process algebras.

Historically the functional verification and the evaluation of performance of an application have been considered as two distinct steps of the system development and verification process : each steps had its own moddel and associated verification techniques. In the last fifteen years instead we have seen the flourishing of a discipline that aims at taking simultaneously into consideration both aspects and that is often referred to as probabilistic verification or, more appropriately, of verification of probabilistic systems. The moving force of the discipline is the need of being able to evaluate the probability of a property expressed as a logic formula. To show why this is an important need, we recall a classical example from system reliability.

Chapitre rédigé par S. DONATELLI et S. HADDAD.

Consider a system whose states can be partitioned in three classes : W, the states in which the system works properly, D, the states in which the system is still working, although in a degraded mode, and F, the states in which the system is not working (failure states). The system can evolve from W states to D or F states, and from D to F states. A classical reliability measure for such a system is the probability of being in a F state within a given time interval I. calssical performance and reliability methods can be easily applied to compute such probability.

If instead we ask for a slightly more refined question, as the probability of failing within I, *given that the system has not passed through a degraded mode of operation*, then we need to express and compute the probability of reaching an F state within I, passing only through W states. A temporal logic (as CSL for example) has temporal operators that allow a simple, and semantically well-founded definition for the above property. In this particular case the formula is : $P_{\leq p}(W\ \mathcal{U}^I F)$ where p is the upper limit of the probability of such an event as fixed by the designer.

This chapter presents the two main themes of probabilistic verification : the temporal logics to express probabilistic verification properties and the techniques to verify such properties for Markov chains.

The first part of the chapter recalls the basic elements of stochastic processes and Markov chains, the second part is devoted to the quantitative verification of discrete time Markov chains, followed by the quantitative verification of continous time Markov chains. The chapter concludes with an overview of the literature on the various techniques for probabilistic verification as well as on a number of extensions to the basic temporal logics presented in the chapter.

8.2. Performance evaluation of Markov models

8.2.1. *A stochastic model for discrete events systems*

In this section we assume that the reader is familiar with the basic probability concepts. For more details the interested reader may consult [FEL 68, FEL 71, TRI 82]

Notations

– $\Pr(E)$ is the probability of event E, while $\Pr(A \mid B)$ is the probability of A given B.

– The term $almost$, in an expression like *almost everywhere* or *almost surely*, means with probability 1.

– $\mathbb{R}$ (resp. $\mathbb{R}^+, \mathbb{R}^{+*}$) denotes the real numbers (resp. non negative and strictly positive reals). If x is a real, then $\lfloor x \rfloor$ denotes its integer part.

– If $E \subseteq \mathbb{R}$ then $Inf(E)$ (resp. $Sup(E)$) denotes the lower (resp. upper) bound of E.

Given a discrete event systen (DES), its execution is characterized by a sequence, possibly infinite, of events $\{e_1, e_2, \ldots\}$ and associated interval of time between successive events in the sequence. Only the events can change the state of the system.

Formally, the stochastic behaviour of a DES is defined by two families of random variables :

– $X_0, \ldots, X_n, \ldots$ defined over the (discrete) state space of the system, denoted as S. In the following, unless otherwise specified, we assume that S is finite. X_0 is the system initial state and X_n $(n > 0)$ is the state after the n^{th} event. The occurrence of an event does not necessarily modify the state of the system, and therefore X_{n+1} may be equal to X_n.

– $T_0, ..., T_n, ...$ defined over $\mathbb{R}^+$, where T_0 is the time interval before the first event and T_n $(n > 0)$ is the time interval between the n^{th} and the $(n+1)^{th}$ event. please note that this interval may be null (*e.g.* a sequence of assignment instructions can be considered as istantaneous with respect to a complex data base transaction involving some input/output activity).

If the initial distribution of variable x_0 is concentrated on a single state s, we say that the process starts in s (*i.e.* $\Pr(x_0 = s) = 1$).

A priori there is no restriction whatsoever on the two families of random variables, but, for the stochastic processes that we shall study in the following, we assume that a discrete event system cannot execute an infinite number of actions in a finite amount of time. that is to say :

$$\sum_{n=0}^{\infty} t_n = \infty \text{ almost surely} \tag{8.1}$$

The above property allows to define the state of the system at a given time instant. let $n(\tau)$ be the random variable defined by :

$$n(\tau) =_{def} min(\{n \mid \sum_{k=0}^{n} t_k > \tau\})$$

according to equation (8.1), $n(\tau)$ is defined *almost everywhere*. As exemplified in figure 8.1, $n(\tau)$ can have jumps of size bigger than one. The state $y(\tau)$ of the system at time τ, is then simply $x_{n(\tau)}$. it is important to remark that $y(\tau)$ it is not equivalent to the stochastic process, but it allows, in most cases, to apply standard solution methods.

The diagram of figure 8.1 represents a possible *execution* of the process and shows the interpretation of each random variable defined above. In the execution the process is initially in state s_4, where it stays until, at time τ_0, it moves to state s_6. At time $\tau_0 + \tau_1$, the system visits, in zero time, the states s_3 and s_{12}, ending up in state s_7,

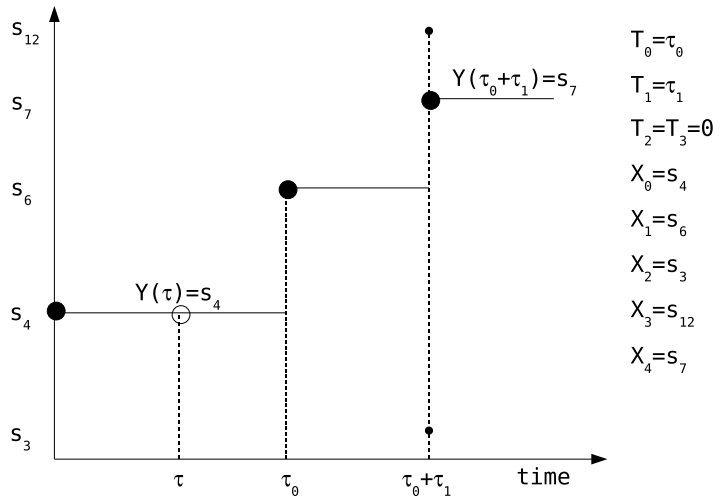

Figure 8.1: an execution of the stochastic process

where it stays for a certain amount of time. The use of $y(\tau)$ in continous time, hides the vanishing states s_3 and s_{12} visited by the process.

The performance evaluation of a discrete event system can be based on two complementary approaches :

– Analysis under transient behaviour, that is to say, the computation of performance measures which are function of the time passed since the start of the system. This type of analysis is well suited for studying the system behaviour in the initialization phase, or for studying systems with final states. Classical applications of transient analysis can be found in the studies aimed at assessing the dependability and reliability of systems [LAP 95, MEY 80, TRI 92].

– Analysis in steady state, that is to say, the computation of performance measures which takes into account only the stationary behaviour of the system, that may be reached after a transient initial phase.

The analysis in steady state makes sense only if such a stationary behaviour exists, a condition that can be expressed as follows, denoting $\boldsymbol{\pi}(\tau)$ the distribution of $y(\tau)$:

$$\lim_{\tau \to \infty} \boldsymbol{\pi}(\tau) = \boldsymbol{\pi} \tag{8.2}$$

where $\boldsymbol{\pi}$ is also a distribution, called the *stationary distribution*.

The transient and stationary distributions are the basis for the computation of *performance indices*. Examples of indices are the steady state probability that a server is up and running, the probability that at time τ a connection has been established, and the mean number of clients waiting for a service. to abstract from the definition of the single performance index, we can introduce the concept of *reward function*, a function f defined on the set of states of the discrete event system and with value onto $\mathbb{R}$. given a distribution $\boldsymbol{\pi}$, the quantity $\sum_{s \in s} \boldsymbol{\pi}(s) \cdot f(s)$ represents the measure of the performance index defined by f.

If f takes values over $\{0, 1\}$, we can consider f as the definition of an *atomic proposition* which is satisfied in state s if $f(s) = 1$ and false otherwise. in the following we shall indicate with $\mathcal{P}$ the set of atomic propositions and with $s \vDash \phi$, with s a state and ϕ an atomic proposition, the fact that s verifies (or satisfies) ϕ. in this context, if $\boldsymbol{\pi}$ is a distribution, the quantity $\sum_{s \vDash \phi} \boldsymbol{\pi}(s)$ represents the measure of the index defined by f.

8.2.2. *Discrete time Markov chains*

Presentation

A Discrete Time Markov Chain (*DTMC*) is a stochastic process with the following characteristics :

– the time interval between the time instants t_n is a constant whose value is 1.

– the next state depends only on the current state, and the transition probability among states remains constant over time[1] :

$$\Pr(X_{n+1} = s_j \mid X_0 = s_{i_0}, ..., X_n = s_i) =$$

$$\Pr(X_{n+1} = s_j \mid X_n = s_i) = p_{ij} =_{def} \mathbf{P}[i, j]$$

and we shall freely mix the two notations p_{ij} and $\mathbf{P}[i, j]$ for the transition probability.

Transient and steady state behaviour of a DTMC

We now recall a number of classical results on the analysis of DTMC : the results will be explained in an intuitive manner, a full mathematical treatment of the topic being out of the scope of this chapter.

The transient analysis is rather simple : the change of state takes place at time instants $\{1, 2, \ldots\}$, and given an initial distribution $\boldsymbol{\pi}_0$ and the transition probability

1. which justifies the definition of *homogeneous* Markov chain

matrix $\mathbf{P}$, we have that $\boldsymbol{\pi}_n$, the distribution of X_n (*i.e.* the state of the chain at time n) can be expressed as $\boldsymbol{\pi}_n = \boldsymbol{\pi}_0 \cdot \mathbf{P}^n$, which is computed using a basic recurrence scheme.

To analyze the asymptotic behaviour of a DTMC we need to investigate a bit further the DTMC behaviour, in particular we shall classify states as follows :

– A state s is said to be *transient* if the probability of visiting s more than once is strictly less than 1. as a consequence, the probability of $\Pr(X_n = s)$ goes to zero as n tends to infinity. A state is said to be *recurrent* if it is not transient.

– A recurrent state s is said to be *null recurrent* if the mean time between two successive visits to s is infinite. Intuitively, a null-recurrent state will be visited at intervals whose mean duration goes to infinity and therefore the probability of visiting s will also tends towards 0.

– A recurrent state s is *not null recurrent* if the mean time between two successive visit to s is finite. If a steady state distribution exists, then it is concentrated on the set of non null recurrent states.

We now explain in detail the steady state analysis procedure for the case of DTMCs with a finite state space. The first step consists in building the following graph :

– the set of nodes is the set of states of the chain ;

– there is an arc from s_i to s_j if $p_{ij} > 0$.

On the graph we compute the strongly connected components (SCC). If an SCC has an exit arc, then all the states of the SCC are transient. All the arcs of a bottom SCC (BSCC), which are components without an exit arc, are non null recurrent. In the particular case of a sink SCC composed by a single state s (*i.e.* $\mathbf{P}[s, s] = 1$), we say that s is an *absorbing* state.

If the graph is strongly connected (there is a single SCC), then the chain is said to be *irreducible*. In the more general case instead each sink SCC constitutes an irreducible subchain.

Even if we consider an irreducible chain, the existence of a steady state distribution is not guaranted. Indeed a chain with two states s_0 and s_1, with an initial distribution concentrated in a single state and transition probabilities $p_{0,1} = p_{1,0} = 1$, keep switching between the two states at each instant of time and therefore it does not converge to any stationary distribution. An irreducible chain is said to be *periodic* of period $k > 1$ if its states can be partitioned into subsets $S_0, S_1, \ldots, S_{k-1}$ such that, from the states in S_i the chain moves, in one step only to states which are in $S_{(i+1) \mod k}$. It is possible to compute the periodicity of a chain with a linear time algorithm (in the size of the graph) that we describe in the following using the graph in 8.2. The algorithm computes first a directed tree that covers all nodes of the chain, using a breadth-first

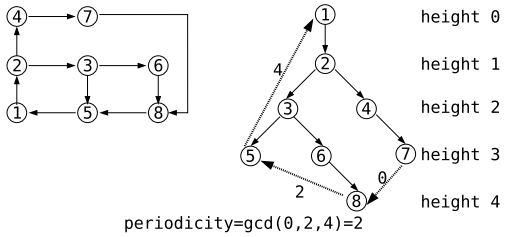

Figure 8.2: Example of the computation of a DTMC periodicity

strategy, that allows to label each node with its "heigth" h. Next steps associate with each arc (u, v) of the graph a weight $w(u, v) = h(u) - h(v) + 1$: as a result all the arcs that are part of the covering tree have a null weight. The periodicity of the graph is then the greatest common divisor (gcd) of the arcs of non null weight. The formal proof of correctness, that we do not develop here is based on the two following observations. Periodicity is the gcd of the length of the elementary circuits of the graphs, and this length is equal to the sum of the weight of the arcs of the circuit.

An irreducible, aperiodic chain (also called *ergodic*) has a stationary distribution, and such a distribution is *independent from the initial distribution*. The computation of the steady state distribution is then rather easy, since $\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n \cdot \mathbf{P}$. Taking the limit as n goes to infinity (which is mathematically sound) we get $\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \mathbf{P}$. Moreover $\boldsymbol{\pi}$ is the single distribution which is a solution for :

$$\mathbf{X} = \mathbf{X} \cdot \mathbf{P} \tag{8.3}$$

Please note that an initial distribution which is solution of the above equation, is *invariant* : whatever the instant of time at which the chain is observed the distribution will be equal to the initial distribution. Equation (8.3) can be solved with a direct method, once we add the normalization equation $\mathbf{X} \cdot \mathbf{1}^T = 1$ where $\mathbf{1}^T$ denotes the column vector of all 1. If the size of the system is large, iterative methods are more effective. The simplest one iterates over $\mathbf{X} \leftarrow \mathbf{X} \cdot \mathbf{P}$ [STE 94].

We now consider the more general case, with the single remaining assumption that the BSCC (denoted as $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$) are aperiodic with stationary distribution $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_k\}$. In this case also the chain has a stationary distribution (which now depends on the initial distribution), given by $\boldsymbol{\pi} = \sum_{i=1}^{k} \Pr(\text{of reaching } \mathcal{C}_i) \cdot \boldsymbol{\pi}_i$. To compute the probability of reaching a BSCC we condition on being in a initial state : $\Pr(\text{of reaching } \mathcal{C}_i) = \sum_{s \in S} \boldsymbol{\pi}_0(s) \cdot \boldsymbol{\pi}'_{\mathcal{C}_i}(s)$ where $\boldsymbol{\pi}'_{\mathcal{C}_i}(s) = \Pr(\text{of reaching } \mathcal{C}_i \mid X_0 = s)$. If $\mathbf{P}_{T,T}$ is the submatrix of the transition matrix limited to transient states, and if $\mathbf{P}_{T,i}$ is the submatrix from transient states towards the states of $\mathcal{C}_i$, then $\boldsymbol{\pi}'_{\mathcal{C}_i} =$

$(\sum_{n\geq 0}(\mathbf{P}_{T,T})^n)\cdot\mathbf{P}_{T,i}\cdot\mathbf{1}^T = (\mathbf{I}-\mathbf{P}_{T,T})^{-1}\cdot\mathbf{P}_{T,i}\cdot\mathbf{1}^T$. The first equality is obtained by conditioning on the length of all possible paths that leads to $\mathcal{C}_i$, while the second one is immediate.

8.2.3. *Continous time Markov chain*

Presentation

A Continuous Time Markov Chain (*CTMC*) has the following characteristics :

– the time interval between the time instants T_n is a random variable distributed as a negative exponential, whose rate depends only on the state X_n. That is to say :

$$\Pr(T_n \leq \tau \,|\, X_0 = s_{i_0}, ..., X_n = s_i, T_0 \leq \tau_0, ..., T_{n-1} \leq \tau_{n-1}) =$$
$$\Pr(T_n \leq \tau \,|\, X_n = s_i) = 1 - e^{\lambda_i \cdot \tau}$$

– The next state depends only on the current state, and the transition probabilities remain constant[2] over time :

$$\Pr(X_{n+1} = s_j \,|\, X_0 = s_{i_0}, ..., X_n = s_i, T_0 \leq \tau_0, ..., T_{n-1} \leq \tau_{n-1}) =$$
$$\Pr(X_{n+1} = s_j \,|\, X_n = s_i) = p_{ij} =_{def} \mathbf{P}[i,j]$$

The DTMC defined by $\mathbf{P}$ is called *embedded chain*. It observes the change of state, independently of the time elapsed in the state. A CTMC state is said to be absorbing if it is absorbing in the embedded DTMC.

Transient and stationary behaviour of a CTMC

In a continuous time Markov chain at any time the evolution of a DES is completely determined by its current state, due to the memoryless property of the exponential distribution.

In particular, the process is fully characterized by the initial distribution $\boldsymbol{\pi}(0)$, matrix $\mathbf{P}$ and by the rates λ_i. Let $\boldsymbol{\pi}(\tau)$ be the distribution of $Y(\tau)$ and write $\pi_k(\tau) = \boldsymbol{\pi}(t)(s_k)$. If δ is small enough, the probability of more than one event occurring in the interval τ and $\tau+\delta$ is very small and can be neglected, and the probability of a change from state k to state k' is approximately equal to $\lambda_k \cdot \delta \cdot p_{kk'}$ (by definition of exponential distribution).

$$\pi_k(\tau+\delta) \approx \pi_k(\tau)\cdot(1-\lambda_k\cdot\delta) + \sum_{k'\neq k}\pi_{k'}(\tau)\cdot\lambda_{k'}\cdot\delta\cdot p_{k'k}$$

2. Also in this case we say that the chain is *homogeneous*

From which we derive

$$\frac{\pi_k(\tau+\delta)-\pi_k(\tau)}{\delta} \approx \pi_k(\tau)\cdot(-\lambda_k) + \sum_{k'\neq k} \pi_{k'}(\tau)\cdot\lambda_{k'}\cdot p_{k'k}$$

and finally :

$$\frac{d\pi_k}{d\tau} = \pi_k(\tau)\cdot(-\lambda_k) + \sum_{k'\neq k} \pi_{k'}(\tau)\cdot\lambda_{k'}\cdot p_{k'k}$$

Let us define matrix $\mathbf{Q}$ as : $q_{kk'} = \lambda_k \cdot p_{kk'}$ pour $k \neq k'$ and $q_{kk} = -\lambda_k (= -\sum_{k'\neq k} q_{kk'})$. We can the rewrite the previous equation as :

$$\frac{d\boldsymbol{\pi}}{d\tau} = \boldsymbol{\pi}\cdot\mathbf{Q} \tag{8.4}$$

Matrix $\mathbf{Q}$ is called *infinitesimal generator* of the CTMC.

According to equation (8.4) the infinitesimal generator completely specifies the evolution of the system. Although this equation clearly establish the memoryless property of the CTMC, it does not give any direct mean of computing the transient behaviour of a CTMC. A possible method, called *uniformisation*, has been defined in [JEN 53], and it is based upon the construction of a second Markov chain which is equivalent to the first one from a probabilistic point of view. This chain is built as follows. Let's choose a value $\mu \geq Sup(\{\lambda_i\})$, and assume that this is the parameter of the exponential distribution of the time until the next change of state, whatever the current state is (from which the term *uniform*). The change of state is defined by the transition matrix $\mathbf{P}^\mu$ defined by : $\forall i \neq j, \mathbf{P}^\mu[s_i, s_j] = (\mu)^{-1}\cdot\lambda_i\cdot\mathbf{P}[s_i, s_j]$. The computation of the infinitesimal generator of such a chain shows immediately that it is equal to the infinitesimal generator of the first CTMC, which implies that, if we disregard transitions, the two CTMCs describe the same stochastic process. We can then compute the transient distribution $\boldsymbol{\pi}(\tau)$ as follows. We first compute the probability of being in state s at time τ, knowing that there have been n changes of state in the interval $[0,\tau]$. This probability can be computed through the embedded Markov chain, and precisely as $\boldsymbol{\pi}(0)\cdot(\mathbf{P}^\mu)^n$. We can then "condition" it through the probability of having n changes of state, knowing that the time between two successive changes follows an exponential distribution. This probability is given by $e^{-\mu\cdot\tau}\cdot(\mu\cdot\tau)^n/n!$, from which we obtain :

$$\boldsymbol{\pi}(\tau) = \boldsymbol{\pi}(0)\cdot(e^{-\mu\cdot\tau}\sum_{n\geq 0}\frac{(\mu\cdot\tau)^n(\mathbf{P}^\mu)^n}{n!})$$

Although there is an infinite sum, in practice the sum converges rather quickly, and the sum can be stopped once the precision required is greater than $e^{-\mu\cdot\tau}\cdot(\mu\cdot\tau)^n/n!$.

We now consider the asymptotic behaviour of a CTMC. Again, the simplest way is to study the embedded chain, which, as observed when explaining uniformization, it is not unique. Let us build a DTMC as follows. Choose $\mu > Sup(\{\lambda_i\})$, since the inequality is strict, it is true that, for each state s, $\mathbf{P}^{\mu}[s,s] > 0$ and therefore each BSCC of this chain is ergodic. As a consequence, a single stationary distribution exists, that measures the steady state probability of the occurrence of a state. Since the uniform chain has the same mean sojourn time in each state, equal to $(1/\mu)$, this also gives the stationary distribution of the CTMC.

In the particular case (rather frequent) in which the embedded chain is ergodic, this distribution can be computed through the solution of the equation $\mathbf{X} = \mathbf{X} \cdot \mathbf{P}^{\mu}$, and $\mathbf{P}^{\mu} = \mathbf{I} + (1/\mu)\mathbf{Q}$. The distribution is therefore the unique solution of the equation :

$$\mathbf{X} \cdot \mathbf{Q} = 0 \quad \text{et} \quad \mathbf{X} \cdot \mathbf{1}^T = 1 \tag{8.5}$$

By analogy, we then say that the CTMC is ergodic.

8.3. Verification of Discrete Time Markov Chain

8.3.1. *Temporal logics for Markov chains*

We consider a « probabilistic » extension of the CTL^* logic, that is named $PCTL^*$. The syntax of this logic is defined inductively upon state formulas and paths formulas.

Définition 1. *Let $\mathcal{P}$ be the set of atomic propositions.*
A $PCTL^*$ state formula (relative to $\mathcal{P}$) is defined by :

E_1 : If $\phi \in \mathcal{P}$ then ϕ is a $PCTL^*$ state formula ;

E_2 : If ϕ and ψ are $PCTL^*$ state formulas then $\neg\phi$ and $\phi \wedge \psi$ are $PCTL^*$ state formulas ;

E_3 : If φ is a $PCTL^*$ path formula, $a \in [0,1]$ is a rational number, and $\bowtie \in \{=, \neq, <, \leq, >, \geq\}$ then $P_{\bowtie a}\varphi$ is a $PCTL^*$ state formula.

A path formula of $PCTL^*$ (relative to $\mathcal{P}$) is defined by :

C_1 : A $PCTL^*$ state formula is a $PCTL^*$ path formula ;

C_2 : if φ and θ are $PCTL^*$ path formulas, then $\neg\varphi$ and $\varphi \wedge \theta$ are $PCTL^*$ path formulas ;

C_3 : If φ and θ are $PCTL^*$ path formulas, then $\mathcal{X}\varphi$ and $\varphi\mathcal{U}\theta$ are $PCTL^*$ path formulas.

MSG: SD : not sure what you mean in the next words **END** Comme dans le cas des systèmes de transitions, deux fragments de cette logique sont particulièrement

intéressants. Two subsets of the $PCTL^*$ formulas are of particular interest. The first subset is called $PCTL$ (by analogy with CTL) and it is built using only the rules E_1, E_2, E_3, C'_3 where C'_3 is defined as « If ϕ and ψ are $PCTL$ state formulas, the $\mathcal{X}\phi$ and $\phi\mathcal{U}\psi$ are $PCTL$ path formulas ». The second subset is called $PLTL$ (by analogy with LTL) and it is built only on the rules E_1, E_3, C'_1, C_2, C_3 where C'_1 is « If $\varphi \in \mathcal{P}$ the φ is a $PLTL$ state formula ».

We now explain how to evaluate the truth value of a $PCTL$, $PLTL$, or $PCTL^*$ formula.

The semantics of formulas is given in the following. We consider a Markov chain $\mathcal{M}$ whose states are labeled by a subset of atomic propositions. We indicate with s a state of the chain and with $\sigma = s_0, s_1, \ldots$ an infinite path in the graph associated to the chain. We denote σ_i the suffix $s_i, s_{i+1}, \ldots$, and $\mathcal{M}, s \models \phi$ the satisfaction of state formula ϕ by state s and $\sigma \models \varphi$ the satisfaction of path formula φ by path σ.

Définition 2. *Let $\mathcal{M}$ be a Markov chain, s a state of the chain, and σ a path of the chain.*
Teh satisfaction of the state formula ϕ by s is inductively defined by :

- *if $\phi \in \mathcal{P}$ then $\mathcal{M}, s \models \phi$ iff s is labelled by ϕ ;*
- *if $\phi \equiv \neg\psi$ then $\mathcal{M}, s \models \phi$ iff $\mathcal{M}, s \not\models \psi$;*
- *$\phi \equiv \psi_1 \wedge \psi_2$ then $\mathcal{M}, s \models \phi$ iff $\mathcal{M}, s \models \psi_1$ and $\mathcal{M}, s \models \psi_2$;*
- *If $\phi \equiv P_{\bowtie a}\varphi$ then $\mathcal{M}, s \models \phi$ iff $\Pr(\{\sigma \models \varphi\} \mid s_0 = s) \bowtie a$.*

The satisfaction of a path formula φ by σ is inductively defined by :

- *If φ is a state formula, then $\sigma \models \varphi$ iff $\mathcal{M}, s_0 \models \phi$;*
- *If $\varphi \equiv \neg\theta$ then $\sigma \models \varphi$ iff $\sigma \not\models \theta$;*
- *If $\varphi \equiv \theta_1 \wedge \theta_2$ then $\sigma \models \varphi$ iff $\sigma \models \theta_1$ and $\sigma \models \theta_2$;*
- *If $\varphi \equiv \mathcal{X}\theta$ then $\sigma \models \varphi$ iff $\sigma_1 \models \theta$;*
- *If $\varphi \equiv \theta_1 \mathcal{U} \theta_2$ then $\sigma \models \varphi$ iff $\exists i\ \sigma_i \models \theta_2$ and $\forall j < i\ \sigma_j \models \theta_1$.*

This semantics assume implicitly that the set of paths that verify a formula is measurable. This hypothesis is justifiable, as can be proved through basic results of measure theory, but this goes beyond the scope of this chapter.

8.3.2. *Verification of $PCTL$ formulas*

Given a DTMC and a $PCTL$ formula ϕ the verification algorithm proceeds by evaluating bottom up the sub-formulas of the syntactic tree of ϕ, from the leaves up to the root. At each step the algorithm evaluates a sub-formula considering as atomic

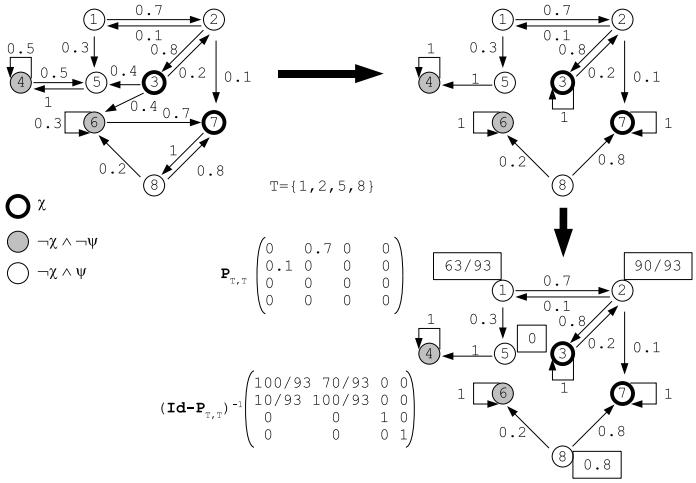

Figure 8.3: Calcul de $P_{\bowtie a}\psi\mathcal{U}\chi$

propositions the operands of the most external operator (of the subformula associated to the tree node considered).

Considering the syntax of $PCTL$ the formulas to be considered are : $\neg\psi, \psi \wedge \chi, P_{\bowtie a}\mathcal{X}\psi, P_{\bowtie a}\psi\mathcal{U}\chi$ where ψ and χ are (formulas transformed into) atomic propositions. We now provide an informal explanaton of the algorithm and its correctness.

$\boxed{\phi = \neg\psi}$ The algorithm labels with ϕ each state not labelled with ψ.

$\boxed{\phi = \psi \wedge \chi}$ The algorithm labels with ϕ each state labelled with ψ et χ.

$\boxed{\phi = P_{\bowtie a}\mathcal{X}\psi}$ The algorithm computes the probability p_s of reaching in a single step a state labelled with ψ, with $p_s \equiv \sum_{s' \models \psi} \mathbf{P}[s, s']$ where $\mathbf{P}$ is the transition matrix of the DTMC. State s is then labelled with ϕ iff $p_s \bowtie a$.

$\boxed{\phi = P_{\bowtie a}\psi\mathcal{U}\chi}$ The algorithm computes the probability of reaching a state labelled by χ, passing only through states labelled by ψ. Let p_s be such a probability. If $s \models \chi$ then $p_s = 1$; if $s \not\models \chi$ and $s \not\models \psi$ then $p_s = 0$. In all other cases, p_s is computed on a transformed DTMC : all the states described above **MSG**: SD : be more precise, if I remember well is $/\!\psi OR\chi$ **END** are made absorbing, and then the probability of reaching χ from s in the new chain. Since each χ state is a BSCC, such a probability can be computed as explained in 8.2.2, and illustrated in figure 8.3. State s is then labelled with ϕ iff $p_s \bowtie a$.

8.3.3. *Aggregation of Markov chains*

In order to establish the correction of the verification algorithm of PLTL, we recall the notions of aggregation in Markov chains. The aggregation of finite Markov chains is an efficient method when one is faced to huge chains [KEM 60]. Its principe is simple : substitute to a chain, an "equivalent" chain where each state of the lumped chain is a set of states of the initial chain. There are different versions of aggregation depending on whether the aggregation is sound for every initial distribution (*strong aggregation*) or for at least one distribution (*weak aggregation*). We simultaneously introduce aggregation for DTMCs and CTMCs. We note π_0 the initial distribution of the chain and X_n (resp. X_t) the random variable describing the state of the DTMC (resp. CTMC) at time n (resp. t) (variables called Y at the beginning of the chapter). $\mathbf{P}$ is the transition matrix of the DTMC and $\mathbf{Q}$ is the infinitesimal generator of the CTMC.

Définition 3. *Let $\mathcal{M}$ be a DTMC (resp. a CTMC) and $\{X_n\}_{n\in\mathbb{N}}$ (resp. $\{X_t\}_{t\in\mathbb{R}^+}$) the family of corresponding random variables. Let $\{S_i\}_{i\in I}$ be a partition of the state space. Define the random variable Y_n for $n \in \mathbb{N}$ (resp. Y_t for $t \in \mathbb{R}^+$) by $Y_n = i$ iff $X_n \in S_i$ (resp. $Y_t = i$ iff $X_t \in S_i$). Then :*

– $\mathbf{P}$ (resp. $\mathbf{Q}$) is strongly lumpable *w.r.t. $\{S_i\}_{i\in I}$*
iff there exists a transition matrix $\mathbf{P}^{lp}$ (resp. an infinitesimal generator $\mathbf{Q}^{lp}$) s.t
$\forall\pi_0$ $\{Y_n\}_{n\in\mathbb{N}}$ (resp. $\{Y_t\}_{t\in\mathbb{R}^+}$) is a DTMC (resp. CTMC)
with transition matrix $\mathbf{P}^{lp}$ (resp. with infinitesimal generator $\mathbf{Q}^{lp}$).

– $\mathbf{P}$ (resp. $\mathbf{Q}$) is weakly lumpable *w.r.t. $\{S_i\}_{i\in I}$*
iff $\exists\pi_0$ $\{Y_n\}_{n\in\mathbb{N}}$ (resp. $\{Y_t\}_{t\in\mathbb{R}^+}$) is a DTMC (resp. CTMC).

While a characterization of the strong aggregation by examination of the transition matrix or the infinitesimal generator is easy, the search of a weak aggregation is much harder [LED 60]. So we introduce exact aggregation, a simple case of weak aggregation.

Définition 4. *Let $\mathcal{M}$ be a DTMC (resp. a CTMC) and $\{X_n\}_{n\in\mathbb{N}}$ (resp. $\{X_t\}_{t\in\mathbb{R}^+}$) the family of corresponding random variables. Let $\{S_i\}_{i\in I}$ be a partition of the state space. Define the random variable Y_n for $n \in \mathbb{N}$ (resp. Y_t for $t \in \mathbb{R}^+$) by $Y_n = i$ iff $X_n \in S_i$ (resp. $Y_t = i$ iff $X_t \in S_i$). Then :*

– A initial distribution π_0 is equiprobable w.r.t. $\{S_i\}_{i\in I}$
if $\forall i \in I, \forall s, s' \in S_i, \pi_0(s) = \pi_0(s')$.

– $\mathbf{P}$ (resp. $\mathbf{Q}$) is exactly lumpable *w.r.t. $\{S_i\}_{i\in I}$*
iff there exists a transition matrix $\mathbf{P}^{lp}$ (resp. an infinitesimal generator $\mathbf{Q}^{lp}$) s.t.
$\forall\pi_0$ equiprobable $\{Y_n\}_{n\in\mathbb{N}}$ (resp. $\{Y_t\}_{t\in\mathbb{R}^+}$) is a DTMC (resp. CTMC)
with transition matrix $\mathbf{P}^{lp}$ (resp. with infinitesimal generator $\mathbf{Q}^{lp}$)
and π_n (resp. π_t) is equiprobable w.r.t. $\{S_i\}_{i\in I}$.

Exact and strong aggregations have simple characterizations [SCH 84] stated in the next proposition.

Proposition 5. *Let $\mathcal{M}$ be a DTMC (resp. a CTMC) and $\mathbf{P}$ (resp. $\mathbf{Q}$) the corresponding transition matrix (resp. the corresponding infinitesimal generator). Then :*

– $\mathbf{P}$ (resp. $\mathbf{Q}$) is strongly lumpable w.r.t. $\{S_i\}_{i\in I}$ iff
$\forall i,j \in I\ \forall s,s' \in S_i\ \sum_{s''\in S_j} \mathbf{P}[s,s''] = \sum_{s''\in S_j} \mathbf{P}[s',s'']$
(resp. $\sum_{s''\in S_j} \mathbf{Q}[s,s''] = \sum_{s''\in S_j} \mathbf{Q}[s',s'']$*)*

– $\mathbf{P}$ (resp. $\mathbf{Q}$) is exactly lumpable w.r.t. $\{S_i\}_{i\in I}$ iff
$\forall i,j \in I\ \forall s,s' \in S_i\ \sum_{s''\in S_j} \mathbf{P}[s'',s] = \sum_{s''\in S_j} \mathbf{P}[s'',s']$
(resp. $\sum_{s''\in S_j} \mathbf{Q}[s'',s] = \sum_{s''\in S_j} \mathbf{Q}[s'',s']$*)*

Proof
We prove the first point and let to the reader the similar proof of the second point.

Assume that the condition is fulfilled, let π_n the distribution of X_n at time n.
Define $\mathbf{P}^{lp}[i,j] = \sum_{s'\in S_j} \mathbf{P}[s,s']$ for an arbitrary $s \in S_i$ (well defined using the condition). Then :
$\sum_{s\in S_i} \pi_{n+1}(s) = \sum_{s\in S_i} \sum_j \sum_{s'\in S_j} \pi_n(s')\mathbf{P}[s',s] =$
$\sum_j \sum_{s'\in S_j} \pi_n(s') \sum_{s\in S_i} \mathbf{P}[s',s] = \sum_j (\sum_{s'\in S_j} \pi_n(s'))\mathbf{P}^{lp}[j,i]$
This établishes that the condition is sufficient.

Assume now that the condition is not fulfilled,
$\exists i,j \in I\ \exists s,s' \in S_i\ \sum_{s''\in S_j} \mathbf{P}[s,s''] \neq \sum_{s''\in S_j} \mathbf{P}[s',s'']$
Let $\pi_{0,s}$ et $\pi_{0,s'}$ be the initial point distributions for s and s'. These two distributions lead to the same Y_0. Then :
$\sum_{s''\in S_j} \pi_{1,s}(s'') = \sum_{s''\in S_j} \mathbf{P}[s,s''] \neq \sum_{s''\in S_j} \mathbf{P}[s',s''] = \sum_{s''\in S_j} \pi_{1,s'}(s')$
This proves that matrix $\mathbf{P}^{lp}$ cannot exist.

◊◊◊

Figure 8.4 illustrates the concept strong aggregation in case of a DTMC.

When the condition of strong aggregation is fulfilled the transition matrix (resp. the infinitesimal generator) of the lumped chain can be directly computed from the transition matrix (resp. from the infinitesimal generator) of the initial chain as stated by the next proposition (immediate consequence of the proof of proposition 5).

Proposition 6. *Let $\mathcal{M}$ be a DTMC (resp. a CTMC) strongly lumpable w.r.t. $\{S_i\}_{i\in I}$. Let $\mathbf{P}^{lp}$ (resp. $\mathbf{Q}^{lp}$) be the transition matrix (resp. the infinitesimal generator) associated with the lumped chain then :*
$\forall i,j \in I, \forall s \in S_i, \mathbf{P}^{lp}[i,j] = \sum_{s'\in S_j} \mathbf{P}[s,s']$ *(resp.* $\mathbf{Q}^{lp}[i,j] = \sum_{s'\in S_j} \mathbf{Q}[s,s']$*)*

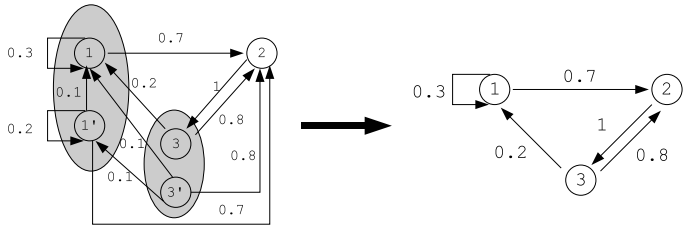

Figure 8.4: An example of strong aggregation in a DTMC

As for strong aggregation, in case of exact aggregation the transition matrix (resp. the infinitesimal generator) of the lumped chain can be directly computed from the transition matrix (resp. from the infinitesimal generator) of the initial chain. Observe that starting with an initial distribution equidistributed over the states of every subset of the partition, at any time the distribution is equidistributed. Consequently, if the DTMC (resp. the CTMC) is ergodic, its stationnary distribution is equidistributed over the states of every subset of the partition. Otherwise stated, knowing the transition matrix (resp. the infinitesimal generator) of the lumped chain, one can compute its stationnary distribution, and deduce (by *local* equidistribution) the stationnary distribution of the initial chain. This last step is impossible with strong aggregation which does not ensure equiprobability of states inside a subset.

Proposition 7. *Let $\mathcal{M}$ be a DTMC (resp. a CTMC) which is exactly lumpable w.r.t. $\{S_i\}_{i\in I}$. Let $\mathbf{P}^{lp}$ (resp. $\mathbf{Q}^{lp}$) be the transition matrix (resp. the infinitesimal generator) associated with the lumped chain, then :*

– $\forall i,j \in I, \forall s \in S_j$ $\mathbf{P}^{lp}[i,j] = (\sum_{s'\in S_i} \mathbf{P}[s',s]) \times (|S_j|/|S_i|)$
(resp. $\mathbf{Q}^{lp}[i,j] = (\sum_{s'\in S_i} \mathbf{Q}[s',s]) \times (|S_j|/|S_i|)$)

– If $\forall i \in I, \forall s,s' \in S_i, \pi_0(s) = \pi_0(s')$ then
$\forall n \in \mathbb{N}$ (resp. $\forall t \in \mathbb{R}^+$), $\forall i \in I, \forall s,s' \in S_i, \pi_n(s) = \pi_n(s')$ (resp. $\pi_t(s) = \pi_t(s')$), where π_n (resp. π_t) is the probability distribution at time n (resp. t)

– If $\mathbf{P}$ (resp. $\mathbf{Q}$) is ergodic and π is its stationnary distribution then
$\forall i \in I, \forall s,s' \in S_i, \pi(s) = \pi(s')$

8.3.4. *Verification of $PLTL$ formulas*

Given a DTMC $\mathcal{M}$ and a $PLTL$ formula ϕ, by definition ϕ is either an atomic proposition, or $P_{\bowtie a}\varphi$ where φ is a path formula built on the operators $\mathcal{X}$, $\mathcal{U}$ and on atomic propositions. The first case is straighforward, while we describe the second case in the following.

As in the previous case, the evaluation proceeds by evaluating the subformulas of φ in the order given by a bottom-up visit of the syntactical tree of the formula. Here after each subformula evaluation transforms both the formula and the DTMC such that at the end the formula becomes an atomic proposition whose evaluation is straightforward. The evaluated subformula φ' is substituted by the atomic proposition $[\varphi']$ in the formula itself.

The transformation of the DTMC is more complex. We descrbe it in the following for the most complex case of a subformula $\varphi' \equiv \psi \mathcal{U} \chi$. Every state s such that $0 < \Pr(\sigma \models \varphi' \mid s_0 = s) < 1$ of the original DTMC is duplicated into s^y, labelled by the propositions labelling s and $[\varphi']$ and s^n labelled by the propositions labelling s. All other states are labelled according to the value of the same probability formula, either 0 or 1. The above probabilities are computed with the same procedure as for $PCTL$. S_o will denote the states that are not duplicated.

The transition probability matrix of the new DTMC is defined as follows :

– The transition probability between states of S_0 is left unchanged as well.

– For all duplicated states, let $py(s) = \Pr(\sigma \models \varphi' \mid s_0 = s)$ and $pn(s) = 1 - py(s)$. The probability to move from a state s' of the original chain to a state s^y (resp. s^n) is the probability of moving from s' to s in the original chain, multiplied by $py(s)$ (resp. $pn(s)$).

– From states s^y (resp. s^n) the chain can only move towards duplicated states s'^y (resp. s'^n) or towards states s' of the original chain such that $py(s') = 1$ (resp. $pn(s') = 1$). The associated transition probabilities are defined by $\mathbf{P'}[s^y, s'^y] = \mathbf{P}[s, s']py(s')/py(s)$ and $P'[s^y, s'] = P[s, s']/py(s)$, similarly for the states s^n.

To complete the definition of the transformed chain we need to define the initial probability of a state s^y (resp. s^n) given that the system starts in state s. This conditional probability is given by $py(s)$ (resp. $pn(s)$). Consequently, $\pi'_0(s^y) = py(s)\pi_0(s)$ et $\pi'_0(s^n) = pn(s)\pi_0(s)$.

Observe that $\mathbf{P'}$ is indeed a transition matrix. We prove it only for a relevant case. $\sum_{s' \in S_o} \mathbf{P'}[s^y, s'] + \sum_{s' \in S \setminus S_o} \mathbf{P'}[s^y, s'^y] =$
$\frac{1}{py(s)} \left(\sum_{s' \in S_o, py(s')=1} \mathbf{P}[s, s'] + \sum_{s' \in S \setminus S_o} \mathbf{P}[s, s']py(s') \right)$
Examining a step of the chain, one observes that the expression between parentheses is the probability $py(s)$.

We show the DTMC transformation caused by subformula $\psi \mathcal{U} \chi$ in figure 8.5.

The correction of this construction is established using the following lemmas. We note $\mathcal{M}'$ the transformed chain. A path is said *normal* if it meets infinitely often S_o.

Lemme 8. *The set of normal paths has measure 1 in $\mathcal{M}$ and in $\mathcal{M}'$.*

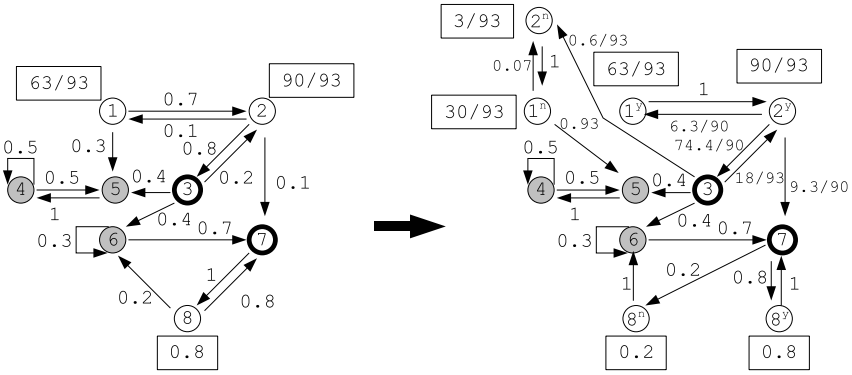

Figure 8.5: CTMC transformation for $PLTL$

Proof
Let us recall that a random path has a probability 1 to meet a sink SCC and to visit infinitely often its states. Examine the different cases of a sink SCC in $\mathcal{M}$ ou $\mathcal{M}'$.

– There exists a state of the SCC fulfilling χ or $\neg\chi \wedge \neg\psi$; this state belonging to S_o will be visited infinitely often.

– All states of the SCC fulfill $\neg\chi \wedge \psi$. In $\mathcal{M}$, this leads to $pn(s) = 1$ for every state s in this SCC. Suppose that in $\mathcal{M}'$ the SCC includes a duplicate state s. Then necessarily there is a path from s to a state s' which fulfills χ. Hence this SCC could not be a sink one.

$\Diamond\Diamond\Diamond$

Let φ'' be a subformula of φ where φ' occurs. Let us note $\varphi''(\varphi' \leftarrow [\varphi'])$, the formula φ'' in which φ' has been substituted by the atomic proposition $[\varphi']$.

Lemme 9. *For every subformula φ'' of φ where φ' occurs, one has for every random path σ of $\mathcal{M}'$,* $\Pr(\sigma \models \varphi''(\varphi' \leftarrow [\varphi']) \Leftrightarrow \varphi'') = 1$

Proof
The base case corresponds à $\varphi'' = \varphi'$ and this is a consequence of the previous lemma since for a normal path σ, $\sigma \models \varphi'$ iff $\sigma \models [\varphi']$. One proves the lemma by induction on the size of the formula observing in the case of temporal operators that a suffix of a normal path is a normal path.

$\Diamond\Diamond\Diamond$

Observe that the previous lemma applies to the case $\varphi'' = \varphi$.

Notations. Define the abstraction mapping abs from states of $\mathcal{M}'$ s.t. $abs(s^y) = abs(s^n) = s$ et $abs(s) = s$ for every $s \in S_o$. Define the stochastic process $\mathcal{M}^{abs}$ whose state space is the one of $\mathcal{M}$ obtained by the abstraction abs applied on $\mathcal{M}'$. The following lemma is the key point for the correction of the algorithm.

Lemme 10. *The stochastic process $\mathcal{M}^{abs}$ is a weak aggregation of the process $\mathcal{M}'$ (w.r.t. the initial distribution π'_0) and it is identical to the Markov chain $\mathcal{M}$.*

Proof
Let us note π_n (resp. π'_n) the distribution of $\mathcal{M}$ (resp. $\mathcal{M}'$) at time n. We prove by recurrence on n that :
$\forall s \in S_o\, \pi_n(s) = \pi'_n(s)$ et $\forall s \in S \setminus S_o\, \pi'_n(s^y) = \pi_n(s)py(s) \wedge \pi'_n(s^n) = \pi_n(s)pn(s)$

For $n = 0$, this is due to the definition of π'_0. Assume that the equations are fulfilled for n. Let us prove it for $n+1$. We only handle the case of a state s^y and let to the reader the other cases.
$\pi'_{n+1}(s^y) = \sum_{s' \in S_o} \pi'_n(s')\mathbf{P}'[s, s^y] + \sum_{s'^y \in S_o} \pi'_n(s'^y)\mathbf{P}'[s'^y, s^y]$
$= \sum_{s' \in S_o} \pi_n(s')\mathbf{P}[s', s]py(s) + \sum_{s'^y | s' \in S \setminus S_o} \pi_n(s')py(s')\mathbf{P}'[s', s]\frac{py(s)}{py(s')}$
$= py(s)\left(\sum_{s' \in S_o} \pi_n(s')\mathbf{P}[s', s] + \sum_{s' \in S \setminus S_o} \pi_n(s')\mathbf{P}'[s', s]\right) = py(s)\pi_{n+1}(s)$

The resultat is then immediate since in $\mathcal{M}^{abs}$, $\forall s \in S \setminus S_o\, \pi^{abs}_n(s) = \pi'_n(s^y) + \pi'_n(s^n)$.

$\Diamond\Diamond\Diamond$

We establish now the correction of the algorithm.

Théorème 11. *Let σ (resp. σ') be a random path of $\mathcal{M}$ (resp. $\mathcal{M}'$). Then :*

$$\Pr_{\mathcal{M}}(\sigma \models \varphi) = \Pr_{\mathcal{M}'}(\sigma' \models \varphi(\varphi' \leftarrow [\varphi']))$$

Proof
$\Pr_{\mathcal{M}}(\sigma \models \varphi) = \Pr_{\mathcal{M}^{abs}}(\sigma^{abs} \models \varphi)$
(lemma 10)
$= \Pr_{\mathcal{M}'}(\sigma' \models \varphi)$
Indeed the truth value of φ for a path σ' depends only on its abstraction σ^{abs}.
$= \Pr_{\mathcal{M}'}(\sigma' \models \varphi(\varphi' \leftarrow [\varphi']))$
(lemma 9)

$\Diamond\Diamond\Diamond$

8.3.5. *Verification of* $PCTL^*$

Given a DTMC and a formula ϕ of $PCTL^*$, the verification algorithm proceeds again through a bottom-up visit of the syntactical tree of the formula ϕ by evaluating the subtrees of ϕ that correspond to $PLTL$ formulas, substituting each verified subformula with an atomic proposition. In each step of the algorithm what needs to be evaluated is a formula of $PLTL$.

8.4. Verification of Continuous Time Markov Chain

Performance evaluation of systems is usually defined in a continuous context. We open this section with a discussion on the limits of classical performance indices, that justify the introduction of a temporal logics for performance evaluation.

8.4.1. *Limitations of standard performance indices*

The classical performance evaluation indices, recalled in section 8.2.1, provide a set of important informations to a system designer, but they do not capture all performance aspects of a system. As an example we consider some performance indices aimed at assessing the dependability of a system.

– *Instantaneous availability* is related to transient behaviour : it represents the probability at time τ of service availability.

– *Steady-state availability* is related to steady-state behaviour : it is represents the probability of service availability in steady-state.

– *Interval availability* : it represents the probability of having the service always available between time τ and τ'.

– *Steady-state interval availability* : it is the steady-state probability that the service is continuously available between two instants of time. Because we are considering the steady-state behaviour, such probability does not depend on the specific points in time, but only on the duration of the interval limited by the two points.

– *Steady-state simultaneous availability and reactivity* : it is the steady-state probability that, upon a request, the system is continuously working until the service is completed and the response time does not exceed a predefined threshold.

While the first two properties can be directly and easily computed from the transient and steady-state probabilities, the computation of the other properties is more involved. It is feasible to devise, for each property, an ad-hoc computation for the probability of interest, but it is more convenient to define a general logics that can express complex performance properties, and for which a general algorithm can be designed.

8.4.2. *A temporal logics for continuous time Markov chains*

The temporal logics CSL ("Continuous Stochastic Logic") that we are going to define is an adaptation of the CTL logics ("Computation Tree Logic" [EME 80]) to CTMC. The logics allows to express formulas that *evaluates over states*, and that are built with the following syntax (in the definition we follow the approach proposed in [BAI 03a]).

Définition 12. *A CSL formula is inductively defined by :*

– If $\phi \in \mathcal{P}$ then ϕ is a CSL formula ;

– If ϕ et ψ are CSL formula then $\neg\phi$ and $\phi \wedge \psi$ are CSL formulas ;

– If ϕ is a CSL formula, $a \in [0,1]$ is a real number, $\bowtie \in \{<, \leq, >, \geq\}$ then $S_{\bowtie a}\phi$ is a CSL formula ;

– If ϕ and ψ are CSL formulas, $a \in [0,1]$ is a real number, $\bowtie \in \{<, \leq, >, \geq\}$ and I is an interval of $\mathbb{R}_{\geq 0}$ then $P_{\bowtie a}\mathcal{X}^I\phi$ and $P_{\bowtie a}\phi\mathcal{U}^I\psi$ are CSL formulas.

The first two definitions are standard CTL formulas, and we do not explain them here in more details. The formula $S_{\bowtie a}\phi$ is satisfied by a state s of the CTMC if, given that the initial state of the chain is s, the cumulative steady-state probability p of the states that satisfy ϕ, verifies $p \bowtie a$. This evaluation is well-defined, since, in a finite CTMC, a steady-state distribution always exists. If the CTMC is ergodic the evaluation of the formula does not depend on the specific state s.

An execution of a stochastic process satisfies $\mathcal{X}^I\phi$ if the first change of state takes place within the interval I and leads to a state that verifies ϕ. A state s satisfies $P_{\bowtie a}\mathcal{X}^I\phi$ if the probability p of the executions of the stochastic process that start in s and satisfy $\mathcal{X}^I\phi$ verifies $p \bowtie a$.

An execution of a stochastic process satisfies $\phi\mathcal{U}^I\psi$ if it exists a time instant $\tau \in I$ such that ψ is true at τ and for all preceeding time instants ϕ is true. A state s satisfies $P_{\bowtie a}\phi\mathcal{U}^I\psi$ if the probability p of the executions that starts in s and satisfy $\phi\mathcal{U}^I\psi$ verifies $p \bowtie a$.

Using CSL, the availability and dependability properties informally defined before can be expressed in more formal terms as :

– *Instantaneous availability* guarantee of 99% :

$$P_{\geq 0.99} true \mathcal{U}^{[\tau,\tau]} disp$$

where $disp$ is an atomic proposition that indicates that the service is available.

– *Steady-state availability* guarantee of 99% :

$$S_{\geq 0.99} disp$$

– *Interval availability* guarantee of 99% :

$$P_{<0.01} true \mathcal{U}^{[\tau,\tau']} \neg disp$$

– *Steady-state interval availability* guarantee of 99% :

$$S_{<0.01} true \mathcal{U}^{[\tau,\tau']} \neg disp$$

– *Steady-state simultaneous availability and reactivity* guarantee of 99% with latency of at most 3 time units :

$$S_{\geq 0.99}(req \Rightarrow P_{\geq 0.99}(disp \mathcal{U}^{[0,3]} ack))$$

where req is the atomic proposition that indicates that a request has been received, and ack is an atomic proposition that indicates that the service has been delivered. Note that the two 99% requirements do not have the same meaning. The condition on the internal operator is a condition on the executions that starts in a particular state, while the condition on the outer operator is a global requirement on all the states of the chain, weighted by their steady-state probabilities.

8.4.3. *Verification algorithm*

Given a CTMC and a CSL formula ϕ, the algorithm evaluates the formula starting from the inner formulas and proceeding from inner to outer formulas, following bottom-up the syntactical tree of the formula ϕ and labelling each state with the subformulas satisfied in that state. At each step, the algorithm evaluates a formula by considering as atomic propositions the operands of the most external operator. The algorithm can be therefore explained considering one operator at a time.

$\boxed{\phi = \neg\psi}$ The algorithm labels with ϕ each state which is not labelled with ψ.

$\boxed{\phi = \psi \wedge \chi}$ The algorithm labels with ϕ every state labelled with both ψ and χ.

$\boxed{\phi = S_{\bowtie a}\psi}$ The algorithm computes the steady state distribution of the CTMC with initial probability concentrated in s (the stochastic process starts in s) as explained in section 8.2.3). The probability of all states labelled with ψ are then summed up and the algorithm labels with ϕ the state s if the sum, let it be p, verifies $p \bowtie a$. Note that for all the states of a BSCC a single computation is needed : indeed either all states of the BSCC satisfy ϕ or none of them does. Similarly, if the CTMC has a single stationary distribution, then the truth value of the formula does not depend on the state.

$\boxed{\phi = P_{\bowtie a}\mathcal{X}^I\psi}$ The occurrence of a transition in a state s in within the interval I and the fact that the state reached upon the transition satisfies ψ are two independent events, and therefore the probability of the paths that satisfy the formula can be computed as the product of the probabilities of the two events. Let $I = [\tau, \tau']$; we assume

a closed interval, without loss of generality (since we are in a continuous domain the fact of including or not the bounds of the interval in the computation does not influence the result). Let $\mathbf{Q}$ the infinitesimal generator of the CTMC, and $\mathbf{P}$ the matrix of the embedded DTMC. The probability of the first event is $e^{\tau\mathbf{Q}[s,s]} - e^{\tau'\mathbf{Q}[s,s]}$, while the probability of the second even is $\sum_{s' \vDash \psi} \mathbf{P}[s, s']$.

$\boxed{\phi = P_{\bowtie a}\psi\mathcal{U}^I\chi}$ The evaluation of this formula requires transient analysis of a CTMC obtained from the original CTMC by some simple transformations. If X is a CTMC, then we shall indicate with X^ϕ the chain obtained by making absorbing all states of X that verify ϕ. In order to simplify the presentation, we consider as separate cases the various type of intervals.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[0,\infty[}\chi$. In this case the executions of the chain on which we cumulate the probability should never leave the states that verify ψ, until a state that verifies χ is reached, without any contraint in time. temps. In other words, we are interested in the behaviour of the chain from its initial state until it enters a state that satisfies $\neg\psi \vee \chi$. Let's consider the chain $X^{\neg\psi\vee\chi}$. If a BSCC of this chain contains a state that verifies χ then the probability that we are interested in is 1 for all states of the BSCC (since all states of a BSCC are recurrent), if no such a state exists in the BSCC, then the probability is 0. Let's call "good" a BSCC associated with a probability 1. This probability only depend on the embedded chain of $X^{\neg\psi\vee\chi}$ and its computation has already been described in section 8.2.2.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[0,\tau]}\chi$. In this case the execution of the procees must visit only states that verify ψ until a state that satisfies χ s reached, and this event should happen at time τ at the latest. In other words, the probability is cumulated along the paths until a state that verifies $\neg\psi \vee \chi$ is reached. We need therefore to compute the following probability $\Pr(X^{\neg\psi\vee\chi}(\tau) \vDash \chi \mid X^{\neg\psi\vee\chi}(0) = s)$.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\tau]}\chi$. In this case the excution of the process must stay in within states that verify ψ during the interval $[0, \tau]$ and it must verify χ at time τ. The case of a change of state at τ is not considered since the probability of this event is zero. The probability to be computed is equal to $\Pr(X^{\neg\psi}(\tau) \vDash \psi \wedge \chi \mid X^{\neg\psi}(0) = s)$.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\infty[}\chi$. In this case the execution of the process must stay in within states that verify ψ during the interval $[0, \tau]$ and then starting from the state s reached at time τ it must verify the furmula $\psi\mathcal{U}^{[0,\infty[}\chi$. The probability to be computed is therefore $\sum_{s' \vDash \psi} \Pr(X^{\neg\psi}(\tau) = s' \mid X^{\neg\psi}(0) = s) \cdot \boldsymbol{\pi}(s')$ where $\boldsymbol{\pi}(s')$ is computed using the procedure for the first case.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\tau']}\chi$. A similar reasoning as for the previous case leads to the following formula :
$\sum_{s' \vDash \psi} \Pr(X^{\neg\psi}(\tau) = s' \mid X^{\neg\psi}(0) = s) \cdot \Pr(X^{\neg\psi\vee\chi}(\tau' - \tau) \vDash \chi \mid X^{\neg\psi\vee\chi}(0) = s')$

8.5. State of the art in the quantitative evaluation of Markov chains

The field of Markov chain verification has started on the verification of DTMCs. The first approach for the verification of LTL over DTMCs (proposed in [VAR 85]) is conceptually very simple : the formula is translated into a Büchi automata, the non-determinism is then removed and a Rabin automata is produced. The synchronized product of this automata with the DTMC produces another DTMC, for which, using a variation of the technique explained in section 8.2, it is possible to compute the required probability. The complexity of the computation is doubly exponential in the size of the formula. An improvement in complexity is given by the algorithm in [COU 95] : a new DTMC is built iteratively from the initial DTMC, and the iteration is driven by the operators of the formula. This is the algorithm that we have presented in section 8.3.4. The resulting algorithm is exponential in the size of the formula, and the authors show that the algorithm has optimal complexity. A third algorithm, proposed in [COU 03], also translates the formula into a Büchi automata. Due to the particular construction followed by the algorithm, it is then possible to compute the probability associated to the formula directly on the synchronized product of the automata and of the formula. This algorithm has an optimal complexity as well, and moreover it provides better performance than the previous one in many practical cases.

A classical technique for evalauting the performance of a system consists in associating "rewards" with states and/or transitions of the chain, and in computing the mean reward or the cumulated reward at time t. Rewards are taken into account by the $PRCTL$ logics, which has been defined in [AND 03], where an evaluation algorithm is also presented.

The first relevant work on the verification of CTMCs has appeared in [AZI 96, AZI 00], where it is shown that CSL verification is decidable. The verification algorithm is extremely complex, since it does not perform the implicit approximations that we have done in the CSL verification algorithm presented in this chapter.

We should remark that verification algorithm may become impractical for large Markov chains. A possible way to solve the problem is to take advantage of a modular specification of the system, substituting a module with a smaller one, which is nevertheless equivalent with respect to the verification of the given formula. This approach has been introduced first in [BAI 03a], and it has been later generalized in [BAI 03b], where various definitions of equivalence are considered.

The CSL logics that was introduced in section 8.4.2 has two main limitation. On one side, the path formulas are defined only in terms of atomic propositions associated to states, and not also in terms of the actions/transitions in the path. On the other side the temporal constraints on path formulas are bound to be intervals, which generates a number of limitations to the expressivity of the temporal constraints in the formula. The first limitation has been eliminated in [BAI 04] : the $asCSL$ logics substitutes to

the temporal operators, a regular expression over states and actions. A different approach is presented instead in [DON 07] : the CSL^{TA} logics there introduced defines the formulas with the support of a one-clock, deterministic, timed automata. CSL^{TA} strictly extends CSL, and it is at least as expressive as $asCSL$. Moreover the verification algorithm is not based on the construction of a number of modified CTMCs, but on the definition of a Markov renewal process, and on the computation of the discrete embedded Markov chain of the process.

A totally different approach to limit the complexity of the verification task has been proposed in [YOU 06]. If $P_{\leq a}\phi$ is the formula to be verified, we can generate a number of random executions, and we can then compute the percentage of the executions that do satisfy ϕ; according to standard probability results, this percentage tends to the probability to be computed. This method is very efficient when the verification of the formula requires only executions that have an upper bound in time.

8.6. Bibliographie

[AND 03] ANDOVA S., HERMANNS H., KATOEN J.-P., « Discrete-time rewards model-checked », *Formal Modelling and Analysis of Timed Systems (FORMATS 2003)*, n° 2791LNCS, Marseille, France, Springer Verlag, p. 88 - 103, 2003.

[AZI 96] AZIZ A., SANWAL K., V.SINGHAL, BRAYTON R., « Verifying continuous-time Markov chains », *8th Int. Conf. on Computer Aided Verification (CAV'96)*, n° 1102LNCS, New Brunswick, NJ, USA, Springer Verlag, p. 269–276, 1996.

[AZI 00] AZIZ A., SANWAL K., V.SINGHAL, BRAYTON R., « Model Chcking continuous-time Markov chains », *ACM Transactions on Computational Logic*, vol. 1, n°1, p. 162–170, 2000.

[BAI 03a] BAIER C., HAVERKORT B., HERMANNS H., KATOEN J.-P., « Model-Checking Algorithms for Continuous Time Markov Chains », *IEEE Transactions on Software Engineering*, vol. 29, n°7, p. 524–541, juillet 2003.

[BAI 03b] BAIER C., HERMANNS H., KATOEN J.-P., WOLF V., « Comparative branching-time semantics for Markov chains », *Concurrency Theory (CONCUR 2003)*, n° 2761LNCS, Marseille, France, Springer Verlag, p. 492 - 507, 2003.

[BAI 04] BAIER C., CLOTH L., HAVERKORT B., KUNTZ M., SIEGLE M., « Model Checking Action- and State-Labelled Markov Chains », *Proc. DSN'04*, IEEE, p. 701-710, 2004.

[COU 95] COURCOUBETIS C., YANNAKAKIS M., « The complexity of probabilistic verification », *Journal of the ACM*, vol. 42(4), p. 857–907, july 1995.

[COU 03] COUVREUR J.-M., SAHEB N., SUTRE G., « An optimal automata approach to LTL model checking of probabilistic systems », *In Proc. 10th Int. Conf. Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'2003)*, n° 2850LNAI, Almaty, Kazakhstan, Springer Verlag, p. 361-375, september 2003.

[DON 07] DONATELLI S., HADDAD S., SPROSTON J., « CSL^{TA} : an Expressive Logic for Continuous-Time Markov Chains », *Proceedings of the 4th International Conference on*

Quantitative Evaluation of Systems (QEST'07), Edinburgh, Scotland, IEEE Computer Society Press, p. 31-40, septembre 2007.

[EME 80] EMERSON E. A., CLARKE E. M., « Characterizing Correctness Properties of Parallel Programs Using Fixpoints », *7^{th} International Colloquium on Automata, Languages and Programming, (ICALP)*, Noordweijkerhout, The Netherland, p. 169-181, 1980.

[FEL 68] FELLER W., *An introduction to probability theory and its applications. Volume I*, John Wiley & Sons, 1968, (third edition).

[FEL 71] FELLER W., *An introduction to probability theory and its applications. Volume II*, John Wiley & Sons, 1971, (second edition).

[JEN 53] JENSEN A., « Markov chains as an aid in the study of Markov processes », *Skand. Aktuarietidskrift*, vol. 3, p. 87–91, 1953.

[KEM 60] KEMENY J., SNELL J., *Finite Markov Chains*, D. Van Nostrand-Reinhold, New York, NY, 1960.

[LAP 95] LAPRIE J., Ed., *Guide de la sûreté de fonctionnement*, Cépaduès - Éditions, Toulouse, France, 1995.

[LED 60] LEDOUX J., *Weak lumpability of finite Markov chains and positive invariance of cones*, research report INRIA-IRISA n° 2801, 1960.

[MEY 80] MEYER J., « On evaluating the performability of degradable computing systems », *IEEE Transactions on Computers*, vol. 29, n°8, p. 720–731, août 1980.

[SCH 84] SCHWEITZER P. J., « Aggregation Methods for Large Markov Chains », *Proceedings of the International Workshop on Computer Performance and Reliability*, North-Holland, p. 275-286, 1984.

[STE 94] STEWART W. J., *Introduction to the numerical solution of Markov chains*, Princeton University Press, USA, 1994.

[TRI 82] TRIVEDI K. S., *Probability & statistics with reliability, queueing, and computer science applications*, Prentice Hall, Englewood Cliffs, NJ, USA, 1982.

[TRI 92] TRIVEDI K. S., MUPPALA J. K., WOOLE S. P., HAVERKORT B. R., « Composite performance and dependability analysis », *Performance Evaluation*, vol. 14, n°3–4, p. 197–215, février 1992.

[VAR 85] VARDI M., « Automatic Verification of Probabilistic Concurrent Finite-State Programs », *FOCS 1985*, p. 327-338, 1985.

[YOU 06] YOUNES H., KWIATKOWSKA M., NORMAN G., PARKER D., « Numerical vs. Statistical Probabilistic Model Checking », *Int. Journal on Software Tools for Technology Transfer (STTT)*, vol. 8(3), p. 216–228, 2006.