Chapitre 8

Vérification quantitative de chaînes de Markov

8.1. Introduction

La complexité des systèmes matériels et logiciels a mis en évidence l'intérêt de méthodes automatiques afin d'accroître le degré de confiance que les exigences fonctionnelles et de performance de ces systèmes seront satisfaites. Si on souhaite prendre en compte le caractère aléatoire d'un système alors ce système se modélisera sous forme d'un processus stochastique. De plus en vue de maîtriser le processus de validation, ces processus stochastiques sont souvent des chaînes de Markov à temps discret (DTMC) ou à temps continu (CTMC) engendrés par un formalisme de haut niveau comme les réseaux de Petri stochastiques ou les algèbres de processus stochastiques.

Pendant longtemps, la vérification fonctionnelle et l'évaluation de performance d'un système ou d'une application se présentaient comme deux étapes distinctes du processus de développement. Chacune disposait de ses modèles et de ses méthodes. Or depuis une quinzaine d'années, de nombreux travaux se situent à la croisée des deux thématiques et sont regroupées sous l'étiquette de vérification probabiliste ou de manière plus appropriée de vérification de systèmes probabilisés.

Ce nouvel axe de recherche répond au besoin des modélisateurs. Ceux-ci souhaitent par exemple évaluer la probabilité de satisfaction d'une propriété exprimée par une formule logique. Donnons en un exemple classique dans le domaine de la sûreté de fonctionnement.

Chapitre rédigé par S. DONATELLI et S. HADDAD.

Considérons un système dont les états sont partitionnés entre états de fonctionnement normal (appelons-les N), des états de fonctionnement en mode dégradé (appelons-les D) et des états de pannes (appelons-les F). Le système évolue de W vers D ou F et de D vers F. Une mesure classique de sûreté de fonctionnement est la probabilité d'une panne dans un intervalle I donné. Dans ce cas les méthodes d'évaluation standard forunissent les résultats escomptés.

Si maintenant nous nous intéressons à la probabilité d'une panne dans l'intervalle I *mais sans passer par le mode dégradé*, nous devons exprimer (et calculer) la probabilité d'atteindre P dans l'intervalle I, en passant uniquement par des éats de N. Une logique temporelle (comme CSL par exemple) possède des opérateurs temporels qui permettent une description simple et sémantiquement fondée. Dans ce cas précis, la formule sera donnée par : $P_{\leq p}(W\ \mathcal{U}^I F)$ où p est la borne supérieure de probabilité recherchée par le concepteur.

L'objectif de ce chapitre est de présenter au lecteur les deux thèmes de recherche importants liés à cet axe : la définition de logiques temporelles pour les chaînes de Markov et la vérification qu'une chaîne de Markov satisfait une formule de logique temporelle.

La première partie du chapitre est constituée de rappels indispensables sur les processus stochastiques et les chaînes de Markov. Puis la deuxième partie est consacrée à la vérification quantitative des chaînes de Markov à temps discret. La partie suivante introduit la vérification des chaînes de Markov à temps continu. Le chapitre se conclut par un panorama des méthodes de vérification des chaînes de Markov proposées par les chercheurs.

8.2. Evaluation de performance de modèles markoviens

8.2.1. *Un modèle stochastique de systèmes à événements discrets*

Nous supposons connues du lecteur les bases de la théorie des probabilités. Pour plus de détails, on pourra se reporter aux ouvrages suivants : [FOA 98, FOA 02] en français ou [FEL 68, FEL 71, TRI 82] en anglais.

Notations

– $\Pr(E)$ désigne la probabilité d'un événement E et $\Pr(A \mid B)$ la probabilité de A sachant B.

– L'adverbe $presque$, dans des expressions telles que *presque partout* ou *presque sûrement*, signifie pour un ensemble de probabilité 1.

– $\mathbb{R}$ (resp. $\mathbb{R}^+, \mathbb{R}^{+*}$) désigne les réels (resp. les réels non négatifs, strictement positifs). Si x est un réel alors $\lfloor x \rfloor$ désigne sa partie entière.

– Si $E \subseteq \mathbb{R}$ alors $Inf(E)$ (resp. $Sup(E)$) désigne la borne inférieure (resp. supérieure) de E.

Une exécution d'un SED se caractérise par une suite (*a priori* infinie) d'événements $\{e_1, e_2, \ldots\}$ séparés par des intervalles de temps. Seuls les événements peuvent changer l'état du système.

Formellement, le comportement stochastique d'un SED est déterminé par deux familles de variables aléatoires :

– $X_0, \ldots, X_n, \ldots$ à valeurs dans l'espace (discret) des états du système, noté S. Dans la suite sauf mention explicite, nous supposerons que cet espace est fini. X_0 représente l'état initial du système et X_n $(n > 0)$ l'état courant après le $n^{ième}$ événement. L'occurrence d'un événement ne modifie pas nécessairement l'état du système, par conséquent X_{n+1} peut être égal à X_n.

– $T_0, ..., T_n, ...$ à valeurs dans $\mathbb{R}^+$ où T_0 représente l'intervalle de temps avant le premier événement et T_n $(n > 0)$ représente l'intervalle de temps entre le $n^{ième}$ et le $(n+1)^{ième}$ événement. Notons que cet intervalle peut être nul (*e.g.* une suite d'instructions considérées comme instantanées au regard de transactions de base de données avec des entrées/sorties).

Lorsque la distribution initiale X_0 est concentrée en un état s, on dira que le processus démarre en s (*i.e.* $\Pr(X_0 = s) = 1$).

A priori, aucune restriction n'est imposée sur ces familles de variables aléatoires. Cependant, pour les catégories de processus que nous étudierons, un SED ne peut exécuter une infinité d'actions en un temps fini. Autrement dit :

$$\sum_{n=0}^{\infty} T_n = \infty \; presque \; sûrement \tag{8.1}$$

Cette propriété nous autorise à définir l'état du système à tout instant. Soit $N(\tau)$, la variable aléatoire définie par :

$$N(\tau) =_{def} Inf(\{n \mid \sum_{k=0}^{n} T_k > \tau\})$$

D'après l'équation [8.1], $N(\tau)$ est définie *presque partout*. Comme on peut le voir sur la figure 8.1, $N(\tau)$ présente des sauts d'amplitude supérieure à 1. L'état $Y(\tau)$ du système à l'instant τ, est alors simplement $X_{N(\tau)}$. Il est important de noter que $Y(\tau)$ n'est pas équivalent au processus stochastique, mais qu'il permet, dans la plupart des cas, de procéder aux analyses standard. Le schéma de la figure 8.1 présente une

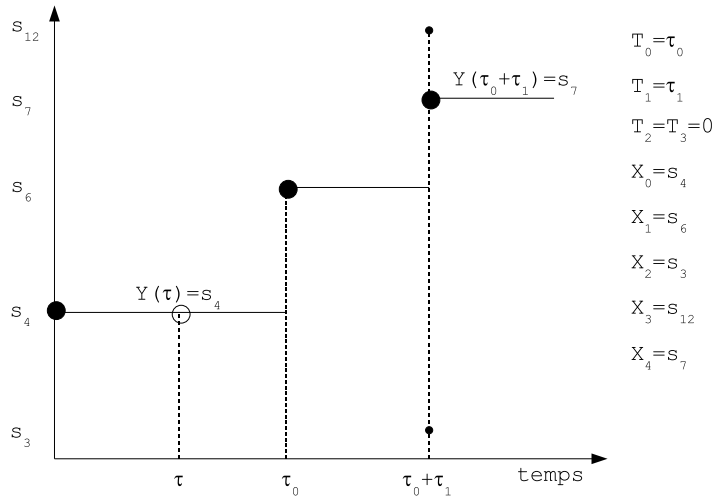

Figure 8.1: Une réalisation du processus stochastique

réalisation possible du processus et illustre l'interprétation de chacune des variables aléatoires introduites plus haut. Dans cet exemple, le processus est initialement dans l'état s_4 et y reste jusqu'à l'instant τ_0 où il passe dans l'état s_6. À l'instant $\tau_0 + \tau_1$, le système visite successivement en un temps nul, les états s_3 et s_{12} avant d'atteindre l'état s_7 où il séjourne un certain temps. L'observation $Y(\tau)$ en temps continu occulte les états évanescents s_3 et s_{12} du processus.

L'évaluation de performance d'un SED conduit à deux types d'analyse :

– L'étude du comportement transitoire , c'est à dire l'obtention de mesures en fonction du temps écoulé depuis l'état initial. Cette étude vise les phases d'initialisation des systèmes et les systèmes à états terminaux. Parmi les domaines d'application, on peut citer l'analyse de fiabilité et de sûreté de fonctionnement [LAP 95, MEY 80, TRI 92].

– L'étude du comportement stationnaire du système. Pour de nombreuses applications, ce qui intéresse le modélisateur est le comportement du système une fois la phase initiale passée, lorsqu'il se stabilise.

Ceci suppose bien entendu qu'un tel comportement stationnaire existe. Ce qui se résume, en notant $\boldsymbol{\pi}(\tau)$ la distribution de $Y(\tau)$, par :

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

où $\boldsymbol{\pi}$ est aussi une distribution, appelée *distribution stationnaire*.

Les distributions transitoires ou stationnaires ne sont qu'un moyen de calculer des *indices de performance*. Par exemple, la probabilité stationnaire qu'un serveur soit opérationnel, la probabilité qu'à l'instant τ une connexion soit établie ou le nombre moyen de clients d'un service sont de tels indices .

Afin de raisonner de manière générique sur les SED on supposera donné dans la suite un ensemble de fonctions définies sur l'ensemble des états et à valeurs dans $\mathbb{R}$. Ainsi une fonction f peut être vue comme un indice de performance et étant donnée une distribution $\boldsymbol{\pi}$, la quantité $\sum_{s \in S} \boldsymbol{\pi}(s) \cdot f(s)$ représente la mesure de cet indice.

Lorsque l'indice est une fonction à valeurs dans $\{0, 1\}$, on peut l'assimiler à une *proposition atomique* satisfaite en un état si la fonction vaut 1. Dans la suite on notera $\mathcal{P}$, l'ensemble des propositions atomiques et $s \vDash \phi$, avec s un état et ϕ une proposition atomique, le fait que s vérifie (ou satisfait) ϕ. Dans ce cas étant donnée une distribution $\boldsymbol{\pi}$, la quantité $\sum_{s \vDash \phi} \boldsymbol{\pi}(s)$ représente la mesure de cet indice.

8.2.2. *Chaînes de Markov à temps discret*

Présentation

Une chaîne de Markov à temps discret (en anglais Discrete Time Markov Chain ou *DTMC*) possède les caractéristiques suivantes :

– L'intervalle de temps entre les instants T_n est une constante de valeur 1

– L'état suivant un état atteint ne dépend que de cet état et les probabilités de transition restent constantes[1] au cours du temps :

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

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

Nous utiliserons indifféremment les deux notations pour les transitions d'état.

Comportement transitoire et stationnaire d'une DTMC

Dans ce paragraphe nous rappelons des résultats classiques en fournissant des justifications intuitives qui ne sauraient constituer des preuves mathématiques.

L'analyse du comportement transitoire ne présente pas de difficulté. Les changements d'état se font aux instants $\{1, 2, \ldots\}$. Etant données une distribution initiale $\boldsymbol{\pi}_0$

1. d'où le terme de chaîne *homogène* utilisé dans les études sur les chaînes de Markov en toute généralité

et la matrice de transition $\mathbf{P}$, alors $\boldsymbol{\pi}_n$ la distribution de X_n (*i.e.* l'état de la chaîne à l'instant n) est donnée par la formule $\boldsymbol{\pi}_n = \boldsymbol{\pi}_0 \cdot \mathbf{P}^n$ qui s'obtient à l'aide d'une récurrence élémentaire.

L'analyse du comportement asympotique des DTMC (pour un ensemble d'états quelconque) conduit à la classification suivante des états :

– Un état s est *transitoire* si la probabilité d'y revenir est inférieure à 1. Par conséquent, sa probabilité d'occurrence $\Pr(X_n = s)$ tend vers 0 lorsque n tend vers l'infini. Un état est appelé *récurrent* s'il n'est pas transitoire.

– Un état est *récurrent nul* si la durée moyenne du retour à cet état est infinie. Intuitivement, une fois atteint, cet état apparaîtra à des intervalles dont la durée moyenne tendra vers l'infini et par conséquent sa probabilité d'occurrence tendra aussi vers 0. Ce raisonnement intuitif est mathématiquement justifié.

– Un état est *récurrent non nul* si la durée moyenne du retour à cet état est finie. Si une distribution stationnaire existe alors elle est concentrée sur les états récurrents non nuls.

Nous allons détailler cette analyse dans le cas d'un espace d'états fini. Considérons le graphe construit de la manière suivante :

– l'ensemble des sommets est l'ensemble des états de la chaîne ;

– il y a un arc de s_i à s_j si $p_{ij} > 0$.

Etudions les composantes fortement connexes (c.f.c.) de ce graphe. Si une c.f.c. a un arc sortant, alors nécessairement, les états de cette c.f.c. sont transitoires. A l'inverse, tous les états d'une c.f.c. puits (*i.e.* sans arc sortant) sont récurrents non nuls. Dans le cas extrême où une c.f.c. puits est réduit à un état s (*i.e.* $\mathbf{P}[s, s] = 1$), on dit que s est un état *absorbant*.

Lorsque le graphe est fortement connexe, la chaîne est dite *irréductible*. Dans le cas général, chaque c.f.c. puits constitue une sous-chaîne irréductible.

Etudions l'existence d'une distribution stationnaire dans le cas d'une chaîne irréductible. Remarquons d'abord qu'elle n'est pas toujours garantie. Ainsi, une chaîne constituée de deux états s_0 et s_1, de distribution initiale concentrée en un état et où $p_{0,1} = p_{1,0} = 1$, alterne entre les deux états et ne converge donc pas vers une distribution stationnaire. En généralisant, une chaîne irréductible est dite *périodique* de période $k > 1$ si on peut partitionner les états en sous-ensembles $S_0, S_1, \ldots, S_{k-1}$ tels que des états de S_i on accède, en un pas, exclusivement aux états de $S_{(i+1) \mod k}$. La périodicité d'une chaîne se détermine par un algorithme en temps linéaire (par rapport à la taille du graphe) dont nous décrivons les principes et que nous illustrons sur la figure 8.2. On construit un arbre orienté couvrant les sommets par un parcours en largeur. Ce parcours en largeur détermine une hauteur des sommets noté h. On affecte un

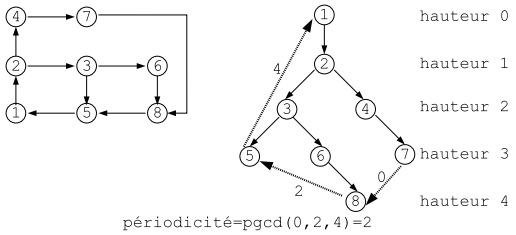

Figure 8.2: Un calcul de périodicité

poids aux arcs du graphe : le poids $w(u,v)$ d'un arc (u,v) est défini par $h(u)-h(v)+1$ ainsi les arcs de l'arbre ont un poids nul. La périodicité du graphe est alors le pgcd des poids non nuls des arcs. Sans entrer dans les détails d'une preuve formelle de cet algorithme, nous indiquons qu'elle repose sur les deux points suivants. La périodicité est le pgcd des longueurs des circuits élémentaires du graphe et la longueur d'un circuit élémentaire est égale à la somme des poids de ces arcs.

Il s'avère qu'une chaîne irréductible et apériodique (dite alors *ergodique*) admet une distribution stationnaire et que celle-ci est *indépendante de la distribution initiale*. Le calcul de cette distribution est relativement facile. En effet, on a $\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n \cdot \mathbf{P}$. En passant à la limite (justifiée), on obtient $\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \mathbf{P}$. De plus, $\boldsymbol{\pi}$ est la seule distribution solution de :

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

Remarquons qu'une distribution initiale, solution de cette équation, est *invariante* : quelque soit l'instant d'observation la distribution courante est identique à la distribution initiale. Afin de résoudre l'équation [8.3], on peut procéder à un calcul direct en complétant par l'équation de normalisation $\mathbf{X} \cdot \mathbf{1}^T = 1$ où $\mathbf{1}^T$ désigne le vecteur colonne composé de 1. Mais les calculs itératifs sont plus intéressants si l'espace d'états est de taille importante. Le plus simple consiste à itérer $\mathbf{X} \leftarrow \mathbf{X} \cdot \mathbf{P}$ [STE 94].

Intéressons-nous maintenant au cas général en supposant uniquement que les c.f.c. puits (notées $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$) sont apériodiques de distributions stationnaires $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_k\}$. Dans ce cas, la chaîne admet aussi une distribution stationnaire (qui cette fois-ci dépend de la distribution initiale). Cette distribution est donnée par la formule $\boldsymbol{\pi} = \sum_{i=1}^{k} \Pr(\text{d'atteindre } \mathcal{C}_i) \cdot \boldsymbol{\pi}_i$. Il reste donc à calculer la probabilité d'atteindre une c.f.c. puits. On évalue cette quantité en partant d'un état fixé puis on la conditionne suivant la distribution initiale : $\Pr(\text{d'atteindre } \mathcal{C}_i) = \sum_{s \in S} \boldsymbol{\pi}_0(s) \cdot \boldsymbol{\pi}'_{\mathcal{C}_i}(s)$ où $\boldsymbol{\pi}'_{\mathcal{C}_i}(s) = \Pr(\text{d'atteindre } \mathcal{C}_i \mid X_0 = s)$. Soit $\mathbf{P}_{T,T}$ la sous-matrice de transition de la chaîne restreinte aux états transitoires et soit $\mathbf{P}_{T,i}$ la sous-matrice de transition de la chaîne des états transitoires vers les états de $\mathcal{C}_i$, alors $\boldsymbol{\pi}'_{\mathcal{C}_i} = (\sum_{n \geq 0} (\mathbf{P}_{T,T})^n) \cdot \mathbf{P}_{T,i} \cdot \mathbf{1}^T =$

$(\mathbf{I}-\mathbf{P}_{T,T})^{-1}\cdot\mathbf{P}_{T,i}\cdot\mathbf{1}^T$. La première égalité s'obtient en conditionnant l'accessibilité de $\mathcal{C}_i$ par la longueur possible du chemin qui y conduit tandis que la seconde se vérifie immédiatement.

8.2.3. *Chaînes de Markov à temps continu*

Présentation

Une chaîne de Markov à temps continu (en anglais Continuous Time Markov Chain ou *CTMC*) a les caractéristiques suivantes :

– L'intervalle de temps entre les instants T_n est une variable aléatoire exponentielle négative dont le taux ne dépend que de l'état X_n. Autrement dit

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

– L'état suivant un état courant ne dépend que de cet état et les probabilités de transition restent constantes[2] au cours du temps :

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

La chaîne discrète définie par la matrice $\mathbf{P}$ est appelée *chaîne incluse*. Elle observe les changements d'état de la CTMC sans tenir compte du temps écoulé. Un état de la CTMC est absorbant s'il est absorbant pour la DTMC incluse.

Comportement transitoire et stationnaire d'une CTMC

Dans les chaînes de Markov à temps continu, en raison de l'absence de mémoire de la loi exponentielle, l'évolution du SED à tout instant est uniquement conditionnée par son état courant.

Plus précisément, le processus se caractérise par sa distribution initiale $\boldsymbol{\pi}(0)$, la matrice $\mathbf{P}$ et les λ_i. Appelons $\boldsymbol{\pi}(\tau)$ la distribution de $Y(\tau)$ et $\pi_k(\tau) = \boldsymbol{\pi}(t)(s_k)$. Si δ est petit, entre τ et $\tau+\delta$ la probabilité de l'occurrence de plus d'un événement est négligeable et la probabilité d'occurrence d'un changement d'état de k à k' est approximativement égale à $\lambda_k \cdot \delta \cdot p_{kk'}$ (par définition de la loi exponentielle).

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

2. ici aussi, on parle de chaîne *homogène*

D'où

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

Et finalement :

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

Définissons la matrice $\mathbf{Q}$ par : $q_{kk'} = \lambda_k \cdot p_{kk'}$ pour $k \neq k'$ et $q_{kk} = -\lambda_k (= -\sum_{k'\neq k} q_{kk'})$. Alors l'équation précédente se réécrit :

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

La matrice $\mathbf{Q}$ est appelée le *générateur infinitésimal* de la CTMC. D'après l'équation [8.4], celui-ci spécifie complètement l'évolution de celle-ci.

Si cette équation établit le caractère sans mémoire d'une CTMC, elle ne fournit pas un moyen pratique de calculer le comportement transitoire de la chaîne. Afin d'y parvenir, nous décrivons une deuxième CTMC équivalente à la première du point de vue probabiliste (une technique introduite dans [JEN 53] et connue sous le nom d'*uniformisation*). Choisissons une valeur $\mu \geq Sup(\{\lambda_i\})$. Quelque soit un état atteint, la durée qui précède le prochain changement d'état suit une loi exponentielle de paramètre (uniforme) μ. Le changement d'état est quant à lui conduit par la matrice de transition $\mathbf{P}^\mu$ définie par $\forall i \neq j, \mathbf{P}^\mu[s_i,s_j] = (\mu)^{-1}\cdot\lambda_i\cdot\mathbf{P}[s_i,s_j]$. Le calcul (immédiat) du générateur infinitésimal de cette deuxième chaine montre qu'il est égal à celui de la première chaîne. On a donc affaire au même processus stochastique si on ne tient pas compte des transitions. La distribution transitoire $\boldsymbol{\pi}(\tau)$ s'obtient de la façon suivante. On calcule la probabilité d'être en s à l'instant τ sachant qu'il y a eu n changements d'états dans l'intervalle $[0,\tau]$. Cette probabilité est donnée par la chaîne incluse et plus précisément par $\boldsymbol{\pi}(0)\cdot(\mathbf{P}^\mu)^n$. Puis on «déconditionne» en calculant la probabilité de n changements sachant que l'intervalle entre deux changements suit la loi exponentielle. Cette probabilité est donnée par $e^{-\mu\cdot\tau}\cdot(\mu\cdot\tau)^n/n!$. On obtient donc :

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

Dans la pratique, la somme infinie ne pose pas de problème car cette somme converge très rapidement et la sommation peut être stoppée dès que la précision requise est supérieure à $e^{-\mu\cdot\tau}\cdot(\mu\cdot\tau)^n/n!$.

Examinons maintenant le comportement asympotique d'une CTMC. La manière la plus simple d'analyser ce comportement consiste à étudier la chaîne incluse. Comme

nous l'avons observé lors de l'approche par uniformisation, celle-ci n'est pas unique. Intéressons-nous à une DTMC obtenue avec un choix de $\mu > Sup(\{\lambda_i\})$. Dans ce cas, tout état s vérifie $\mathbf{P}^{\mu}[s,s] > 0$ et par conséquent chaque c.f.c. puits de cette chaîne est ergodique. Ceci implique qu'elle admet une distribution stationnaire. Cette distribution mesure la probabilité stationnaire d'occurrence d'un état. Mais puisque la description (uniforme) de la chaîne implique un temps de séjour moyen identique dans chacun des états ($1/\mu$), elle nous fournit aussi la distribution stationnaire de la CTMC.

Dans le cas particulier (mais fréquent) où la chaine incluse est ergodique, cette distribution est obtenue par résolution de l'équation $\mathbf{X} = \mathbf{X} \cdot \mathbf{P}^{\mu}$. Nous remarquons que $\mathbf{P}^{\mu} = \mathbf{I} + (1/\mu)\mathbf{Q}$. Donc la distribution est aussi l'unique solution de l'équation :

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

Par analogie, on dit alors que la CTMC est ergodique.

8.3. Vérification de chaînes de Markov à temps discret

8.3.1. *Logiques temporelles pour chaînes de Markov*

Nous introduisons ici une version « probabiliste » de la logique CTL^* que nous désignerons sous le nom de $PCTL^*$. La syntaxe de cette logique est définie inductivement à l'aide de formules d'état et de chemin.

Définition 1. *Soit $\mathcal{P}$, un ensemble de propositions atomiques.*
Une formule d'état de $PCTL^*$ (relative à $\mathcal{P}$) est définie inductivement par :

E_1 : Si $\phi \in \mathcal{P}$ alors ϕ est une formule d'état de $PCTL^*$;

E_2 : Si ϕ et ψ sont des formules d'état de $PCTL^*$ alors $\neg\phi$ et $\phi \wedge \psi$ sont des formules de $PCTL^*$;

E_3 : Si φ est une formule de chemin de $PCTL^*$, $a \in [0,1]$ est un rationnel, $\bowtie \in \{=, \neq, <, \leq, >, \geq\}$ alors $P_{\bowtie a}\varphi$ est une formule d'état de $PCTL^*$.

Une formule de chemin de $PCTL^*$ (relative à $\mathcal{P}$) est définie inductivement par :

C_1 : Une formule d'état de $PCTL^*$ est une formule de chemin ;

C_2 : Si φ et θ sont des formules de chemin de $PCTL^*$ alors $\neg\varphi$ et $\varphi \wedge \theta$ sont des formules de chemin de $PCTL^*$;

C_3 : Si φ et θ sont des formules de chemin de $PCTL^*$ alors $\mathcal{X}\varphi$ et $\varphi \mathcal{U} \theta$ sont des formules de chemin de $PCTL^*$.

Comme dans le cas des systèmes de transitions, deux fragments de cette logique sont particulièrement intéressants. Le premier fragment noté $PCTL$ par analogie avec

CTL est constitué des règles de formation E_1, E_2, E_3, C'_3 où C'_3 s'énonce « Si ϕ et ψ sont des formules d'état de $PCTL$ alors $\mathcal{X}\phi$ et $\phi\mathcal{U}\psi$ sont des formules de chemin de $PCTL$ ». Le deuxième fragment noté $PLTL$ par analogie avec LTL est constitué des règles de formation E_1, E_3, C'_1, C_2, C_3 où C'_1 s'énonce « Si $\varphi \in \mathcal{P}$ alors φ est une formule d'état de $PLTL$ ».

Nous expliquons dans les prochaines sections comment évaluer une formule de $PCTL$, de $PLTL$ et de $PCTL^*$.

La sémantique des formules est donnée ci-dessous. On considère $\mathcal{M}$ une chaîne de Markov dont les états sont étiquetés par un sous-ensemble de propositions atomiques. On note s un état de la chaîne et $\sigma = s_0, s_1, \ldots$ un chemin infini dans le graphe associé à la chaîne de Markov. Le suffixe $s_i, s_{i+1}, \ldots$ est noté σ_i. On note aussi $\mathcal{M}, s \models \phi$ la satisfaction de la formule d'état ϕ par l'état s et $\sigma \models \varphi$ la satisfaction de la formule de chemin φ par le chemin σ.

Définition 2. *Soit $\mathcal{M}$ une chaîne de Markov, s un état de la chaîne et σ un chemin de la chaîne.*
La satisfaction d'une formule d'état ϕ par s est définie inductivement par :

- *Si $\phi \in \mathcal{P}$ alors $\mathcal{M}, s \models \phi$ ssi ϕ étiquette s ;*
- *Si $\phi \equiv \neg\psi$ alors $\mathcal{M}, s \models \phi$ ssi $\mathcal{M}, s \not\models \psi$;*
- *$\phi \equiv \psi_1 \wedge \psi_2$ alors $\mathcal{M}, s \models \phi$ ssi $\mathcal{M}, s \models \psi_1$ et $\mathcal{M}, s \models \psi_2$;*
- *Si $\phi \equiv P_{\bowtie a}\varphi$ alors $\mathcal{M}, s \models \phi$ ssi $\Pr(\{\sigma \models \varphi\} \mid s_0 = s) \bowtie a$.*

La satisfaction d'une formule de chemin φ par σ est définie inductivement par :

- *Si φ est une formule d'état alors $\sigma \models \varphi$ ssi $\mathcal{M}, s_0 \models \phi$;*
- *Si $\varphi \equiv \neg\theta$ alors $\sigma \models \varphi$ ssi $\sigma \not\models \theta$;*
- *Si $\varphi \equiv \theta_1 \wedge \theta_2$ alors $\sigma \models \varphi$ ssi $\sigma \models \theta_1$ et $\sigma \models \theta_2$;*
- *Si $\varphi \equiv \mathcal{X}\theta$ alors $\sigma \models \varphi$ ssi $\sigma_1 \models \theta$;*
- *Si $\varphi \equiv \theta_1 \mathcal{U} \theta_2$ alors $\sigma \models \varphi$ ssi $\exists i\ \sigma_i \models \theta_2$ et $\forall j < i\ \sigma_j \models \theta_1$.*

Cette sémantique suppose implicitement que l'ensemble des chemins qui vérifient une formule est mesurable. Cette supposition est justifiée et se démontre à l'aide de résultats élementaires de la théorie de la mesure mais qui dépassent le cadre de cet ouvrage. Aussi nous ne reviendrons plus sur ce point.

8.3.2. *Vérification de $PCTL$*

Etant données une DTMC et une formule ϕ de PCTL, l'algorithme de vérification procède par évaluation successive des sous-formules de ϕ en «remontant» l'arbre

syntaxique de la formule ϕ des feuilles à la racine et en étiquetant chaque état avec les sous-formules qu'il vérifie. Ainsi chaque étape de l'algorithme évalue une formule en interprétant les opérandes de l'opérateur le plus externe comme des propositions atomiques.

D'après les règles de construction, les formules à considérer sont les suivantes : $\neg\psi, \psi \wedge \chi, P_{\bowtie a}\mathcal{X}\psi, P_{\bowtie a}\psi\mathcal{U}\chi$ où ψ et χ sont des (formules transformées en) propositions atomiques. Nous indiquons ci-dessous comment l'algorithme procède en justifiant (informellement) la correction de l'algorithme.

$\boxed{\phi = \neg\psi}$ L'algorithme étiquette avec ϕ chaque état non étiqueté avec ψ.

$\boxed{\phi = \psi \wedge \chi}$ L'algorithme étiquette avec ϕ chaque état étiqueté avec ψ et χ.

$\boxed{\phi = P_{\bowtie a}\mathcal{X}\psi}$ L'algorithme calcule la probabilité, disons p_s, d'atteindre en un pas un état étiqueté par ψ. Autrement dit, $p_s \equiv \sum_{s' \models \psi} \mathbf{P}[s, s']$ avec $\mathbf{P}$ la matrice de transition de la chaîne. s est alors étiqueté par ϕ ssi $p_s \bowtie a$.

$\boxed{\phi = P_{\bowtie a}\psi\mathcal{U}\chi}$ L'algorithme calcule la probabilité d'atteindre un état étiqueté par χ en passant uniquement par des états étiquetés par ψ. Notons p_s cette probabilité. Si $s \models \chi$ alors $p_s = 1$; si $s \not\models \chi$ et $s \not\models \psi$ alors $p_s = 0$. Afin de calculer p_s dans les autres cas, l'algorithme transforme la chaîne de Markov en rendant absorbant les états décrits précédemment, puis il calcule la probabilité dans cette chaîne d'atteindre les états qui satisfont χ devenus des composantes fortement connexes puits. Le calcul de cette probabilité a été décrit dans la section 8.2.2 et il est illustré par la figure 8.3. s est alors étiqueté par ϕ ssi $p_s \bowtie a$.

8.3.3. *Vérification de* $PLTL$

Etant données une DTMC $\mathcal{M}$ et une formule ϕ de PCTL, on remarque que ϕ est soit une proposition atomique, soit $P_{\bowtie a}\varphi$ où φ est une formule de chemin obtenue à partir des opérateurs $\mathcal{X}$,$\mathcal{U}$ et des propositions atomiques. Dans le premier cas la vérification est triviale. Aussi nous allons détailler le deuxième cas.

Comme précédemment, le principe de cette vérification consiste à évaluer les sous-formules de φ en remontant l'arbre syntaxique de la formule. Cependant, après chaque évaluation, l'algorithme transforme *à la fois la DTMC et la formule* de telle façon que la formule finale est une proposition atomique qui s'évalue immédiatement. La transformation de la formule substitue à une sous-formule φ' une nouvelle proposition atomique notée $[\varphi']$.

La transformation de la DTMC est plus complexe. Nous la décrivons dans le cas le plus difficile où la sous-formule $\varphi' \equiv \psi\mathcal{U}\chi$. Chaque état s t.q. $0 < \Pr(\sigma \models \varphi' \mid$

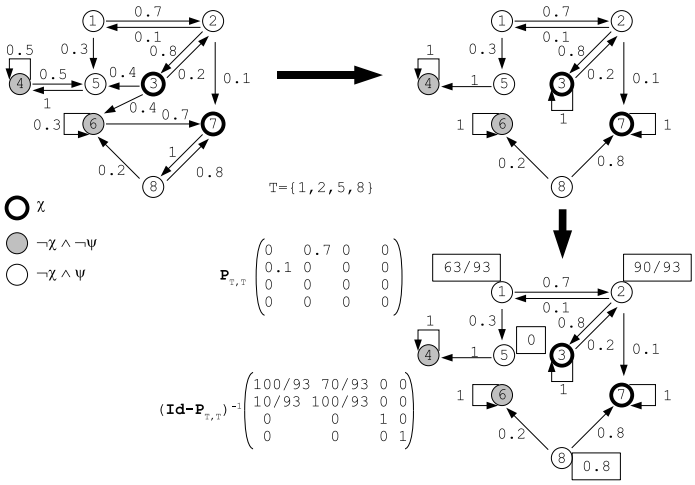

Figure 8.3: Calcul de $P_{\bowtie a}\psi\mathcal{U}\chi$

$s_0 = s) < 1$ (cette probabilité étant calculée dans la DTMC courante, dite chaîne originelle) est dupliqué en deux états s^y étiqueté par $[\varphi']$ et s^n non étiqueté par $[\varphi']$. Les autres états sont étiquetés selon la valeur de cette probabilité (0 or 1).

La spécification des probabilités de transition entre états se fait ainsi :

– Les probabilités entre états de la chaîne originelle sont inchangés.

– Pour les états dupliqués, notons $py(s) = \Pr(\sigma \models \varphi' \mid s_0 = s)$ et $pn(s) = 1 - py(s)$. La probabilité de transition d'un état s' de la chaîne originelle vers s^y (resp. s^n) est celle de s' vers s dans la chaîne originelle multipliée par $py(s)$ (resp. $pn(s)$).

– Il y a uniquement des transitions de s^y (resp. s^n) vers des états s'^y (resp. s'^n) ou vers des états s' de la chaîne originelle t.q. $py(s') = 1$ (resp. $pn(s') = 1$). Les probabilités associées sont définies par $\mathbf{P}'[s^y, s'^y] = \mathbf{P}[s, s']py(s')/py(s)$ et $P'[s^y, s'] = P[s, s']/py(s)$ et de manière similaire pour les états s^n.

Afin de compléter la spécification de la transformation, il faut définir la probabilité de démarrer dans l'état s^y (resp. s^n) sachant que l'on démarre dans l'état s. Cette probabilité conditionnelle est $py(s)$ (resp. $pn(s)$).

Nous illustrons la transformation de la chaîne sur la figure 8.4 pour la sous-formule $\psi\mathcal{U}\chi$. Les probabilités $py(s)$ et $pn(s)$ sont calculées par le même procédé que pour $PCTL$.

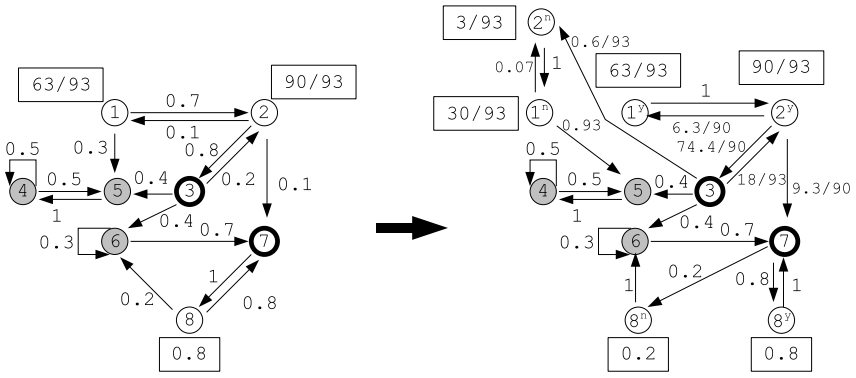

Figure 8.4: Transformation de chaîne pour $PLTL$

La correction de cette construction s'établit à partir des observations suivantes. On note $\mathcal{M}'$ la chaîne transformée. Un chemin est dit *normal* s'il rencontre infiniment souvent les états non dupliqués que l'on notera S_o. D'après la définition d'un état dupliqué, l'ensemble des chemins normaux est de mesure 1 dans $\mathcal{M}$ et dans $\mathcal{M}'$. Aussi dans la nouvelle chaîne, $\Pr(\sigma \models \varphi' \Leftrightarrow \sigma \models [\varphi']) = 1$ puisque cette équivalence est vraie pour les chemins normaux. On démontre par induction que si $\varphi(\varphi' \leftarrow [\varphi'])$ désigne la formule φ dans laquelle φ' a été remplacée par $[\varphi']$ alors $\Pr(\sigma \models \varphi \Leftrightarrow \sigma \models \varphi(\varphi' \leftarrow [\varphi'])) = 1$.

Notations. S_o désigne l'ensemble des états non dupliqués. Soit $\sigma = s_0, s_1, \ldots$ un chemin infini, $\sigma[i,j]$ désigne le chemin fini $s_i, \ldots, s_j$ et $\sigma[i]$ désigne l'état s_i. $Path(s, s')$ avec $s \in S$ et $s' \in S_o$ désigne l'ensemble des chemins finis $s_0, \ldots, s_n$ avec $s_0 = s, s_n = s'$ et $\forall i < n\ s_i \notin S_o$. Remarquons que si $s \in S_o$ alors $Path(s, s)$ se réduit au chemin s et $Path(s, s') = \emptyset$ lorsque $s \neq s'$.

Définissons la fonction d'abstraction abs des états de $\mathcal{M}'$ t.q. $abs(s^y) = abs(s^n) = s$ et $abs(s) = s$ pour tout $s \in S_o$. Cette fonction d'abstraction s'étend aux chemins. Nous affirmons que pour tout chemin fini $\sigma_{fin} = s_0, \ldots, s_n$ de $\mathcal{M}$ t.q. $\sigma_{fin} \in Path(s_0, s_n)$ (et donc $s_n \in S_o$), on a l'égalité :

$$\Pr_{\mathcal{M}'}(\{\sigma \mid abs(\sigma[0,n]) = \sigma_{fin}\} \mid abs(\sigma[0]) = s_0) = \Pr_{\mathcal{M}}(\{\sigma \mid \sigma[0,n] = \sigma_{fin}\} \mid \sigma[0] = s_0) \quad (8.6)$$

Nous avons indicé les probabilités par les chaînes afin d'éviter les ambiguïtés.

Nous prouvons cette égalité dans le cas où $py(s_n) = 1$ (l'autre cas est analogue).

$\Pr_{\mathcal{M}'}(\{\sigma \mid abs(\sigma[0,n]) = \sigma_{fin}\} \mid abs(\sigma[0]) = s_0)$
$= \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma[0,n] = s_0^y, \ldots, s_{n-1}^y, s_n\} \mid abs(\sigma[0]) = s_0)$
$= py(s_0)\mathbf{P}[s_0, s_1]\frac{py(s_1)}{py(s_0)} \ldots \mathbf{P}[s_{n-1}, s_n]\frac{1}{py(s_{n-1})}$

$= \mathbf{P}[s_0, s_1] \dots \mathbf{P}[s_{n-1}, s_n]$
$= \Pr_{\mathcal{M}}(\{\sigma \mid \sigma[0,n] = \sigma_{fin}\} \mid \sigma[0] = s_0)$
On notera cette quantité $\mathbf{P}[\sigma_{fin}]$.

Définissons le processus stochastique $\mathcal{M}^{abs}$ dont l'espace d'états est celui de $\mathcal{M}$ obtenu par l'abstraction abs à partir de $\mathcal{M}'$. Lorsque $\mathcal{M}^{abs}$ démarre dans un état $s_o \in S_0$, elle se comporte comme la chaîne $\mathcal{M}$ démarrée en s_o (pour les spécialistes, il s'agit d'une agrégation faible). Cette affirmation est une conséquence des trois égalités qui se démontrent par récurrence sur n :

$$\forall s \in S_o, \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma[n] = s\} \mid \sigma[0] = s_o) = \Pr_{\mathcal{M}}(\{\sigma \mid \sigma[n] = s\} \mid \sigma[0] = s_o)$$

$$\forall s \in S \setminus S_o, \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma[n] = s^y\} \mid \sigma[0] = s_o) = py(s)\Pr_{\mathcal{M}}(\{\sigma \mid \sigma[n] = s\} \mid \sigma[0] = s_o)$$

$$\forall s \in S \setminus S_o, \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma[n] = s^n\} \mid \sigma[0] = s_o) = pn(s)\Pr_{\mathcal{M}}(\{\sigma \mid \sigma[n] = s\} \mid \sigma[0] = s_o)$$

Nous sommes maintenant en mesure de démontrer la correction de la construction.
$\Pr_{\mathcal{M}}(\{\sigma \mid \sigma \models \varphi\} \mid \sigma[0] = s)$
$= \sum_{s' \in S_o} \sum_{\sigma_{fin} \in Paths_o(s,s')} \mathbf{P}[\sigma_{fin}] \Pr_{\mathcal{M}}(\{\sigma \mid \sigma_{fin}\sigma \models \varphi\} \mid \sigma[0] = s')$
Cette décomposition est valable car elle correspond à une union finie ou dénombrable de sous-ensembles disjoints dont la mesure cumulée est 1.
$= \sum_{s' \in S_o} \sum_{\sigma_{fin} \in Paths_o(s,s')} \mathbf{P}[\sigma_{fin}] \Pr_{\mathcal{M}^{abs}}(\{\sigma \mid \sigma_{fin}\sigma \models \varphi\} \mid \sigma[0] = s')$
D'après le résultat d'agrégation faible.
$= \sum_{s' \in S_o} \sum_{\sigma_{fin} \in Paths_o(s,s')} \mathbf{P}[\sigma_{fin}] \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma_{fin}\sigma \models \varphi\} \mid \sigma[0] = s')$
Car la satisfaction de la formule ne dépend que de l'abstraction.
$= \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma \models \varphi\} \mid abs(\sigma[0]) = s)$
En utilisant l'équation 8.6 et la décomposition appliquée aux chemins de $\mathcal{M}'$.
$= \Pr_{\mathcal{M}'}(\{\sigma \mid \sigma \models \varphi(\varphi' \leftarrow [\varphi'])\} \mid abs(\sigma[0]) = s)$
En raison des observations relatives aux chemins normaux.

8.3.4. *Vérification de* $PCTL^*$

Etant données une DTMC et une formule ϕ de $PCTL^*$, l'algorithme de vérification procède dans l'arbre syntaxique de la formule ϕ par évaluation successive des sous-arbres de ϕ correspondant aux formules de $PLTL$ et en étiquetant chaque état avec les sous-formules qu'il vérifie et en substituant à la sous-formule une proposition atomique. Ainsi chaque étape de l'algorithme évalue une formule de $PLTL$.

8.4. Vérification de chaînes de Markov à temps continu

Les indices de perfomances sont définis le plus souvent dans le cadre du temps continu. Aussi nous commençons cette section par une discussion sur les indices qui conduit à motiver l'usage d'une logique temporelle.

8.4.1. *Limites des indices de performance standard*

Les indices de performance définis lors de la section 8.2.1 apportent des informations précieuses au concepteur du système. Cependant ils ne répondent pas à tous les besoins en terme d'évaluation. Illustrons ce point à l'aide de l'exemple de la disponibilité d'un service. Voici quelques propriétés relatives à ce concept :

– Garantie de disponibilité *instantanée* en régime transitoire. Il s'agit de la probabilité à un instant τ de la disponibilité du service.

– Garantie de disponibilité instantanée en régime stationnaire. Il s'agit de la probabilité à un instant donné de la disponibilité du service en régime stationnaire.

– Garantie de disponibilité *dans la durée* en régime transitoire. Il s'agit de la probabilité que le service soit constamment disponible entre deux instants τ et τ'.

– Garantie de disponibilité dans la durée en régime stationnaire. Il s'agit de la probabilité que le service soit constamment disponible entre deux instants en régime stationnaire. Puisque le processus est en régime stationnaire. Cette mesure ne dépend que la durée de l'intervalle constitué des deux instants.

– Garantie de disponibilité et de temps de réponse en régime stationnaire. Il s'agit la probabilité qu'après une requête, le service soit fonctionnel jusqu'à la réponse et que le temps de réponse n'excède pas une borne donnée.

Si les deux premières propriétés se déduisent facilement des distributions stationnaires et transitoires, il n'en est pas de même des autres. On pourrait imaginer un algorithme ad hoc pour chacune de celles-ci. Mais, il est plus judicieux d'introduire une logique afin d'exprimer des indices de performance complexes et de concevoir un algorithme général d'évaluation de formules de cette logique.

8.4.2. *Une logique temporelle pour les chaînes de Markov à temps continu*

La logique temporelle CSL («Continuous Stochastic Logic») que nous allons détailler est une adaptation de la logique CTL («Computation Tree Logic» [EME 80]) aux chaînes de Markov à temps continu. Elle exprime des formules *qui s'évaluent sur les états* et dont la syntaxe est la suivante. Ici, nous suivrons principalement l'approche de [BAI 03a].

Définition 3. *Une formule de CSL est définie inductivement par :*

– Si $\phi \in \mathcal{P}$ alors ϕ est une formule de CSL ;

– Si ϕ et ψ sont des formules de CSL alors $\neg\phi$ et $\phi \wedge \psi$ sont des formules de CSL ;

– Si ϕ est une formule de CSL, $a \in [0,1]$ est un réel, $\bowtie \in \{<, \leq, >, \geq\}$ alors $S_{\bowtie a}\phi$ est une formule de CSL ;

– Si ϕ et ψ sont des formules de CSL, $a \in [0,1]$ est un réel, $\bowtie \in \{<, \leq, >, \geq\}$ et I est un intervalle de $\mathbb{R}_{\geq 0}$ alors $P_{\bowtie a}\mathcal{X}^I\phi$ et $P_{\bowtie a}\phi\mathcal{U}^I\psi$ sont des formules de CSL.

Seule l'interprétation des deux derniers points nécessite quelques explications. La formule $S_{\bowtie a}\phi$ est satisfaites par s un état de la chaîne si, pour le processus démarré en s, la probabilité stationnaire cumulée (disons p) des états qui satisfont ϕ vérifie $p \bowtie a$. L'évaluation de cette formule est bien définie car pour une CTMC finie, une distribution stationnaire existe toujours. Notons que l'évaluation de cette formule est indépendante de l'état considéré si la chaîne est ergodique.

Une réalisation du processus stochastique satisfait $\mathcal{X}^I\phi$ si le premier changement d'état a lieu dans l'intervalle I et l'état atteint vérifie ϕ. Un état s satisfait $P_{\bowtie a}\mathcal{X}^I\phi$ si la probabilité (disons p) qu'une réalisation du processus démarré en s satisfasse la contrainte énoncée vérifie $p \bowtie a$.

Une réalisation du processus stochastique satisfait $\phi\mathcal{U}^I\psi$ s'il existe un instant $\tau \in I$ tel que ψ soit satisfait et qu'à tous les instants précédents ϕ soit satisfait. Un état s satisfait $P_{\bowtie a}\phi\mathcal{U}^I\psi$ si la probabilité (disons p) qu'une réalisation du processus démarré en s satisfasse la contrainte énoncée vérifie $p \bowtie a$.

A titre d'exemple, nous formalisons maintenant les propriétés de disponibilité énoncées plus haut.

– Garantie de disponibilité instantanée en régime transitoire de 99% :

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

où $disp$ est une proposition atomique indiquant si le service est disponible.

– Garantie de disponibilité instantanée en régime stationnaire de 99% :

$$S_{\geq 0.99} disp$$

– Garantie de disponibilité dans la durée en régime transitoire de 99% :

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

– Garantie de disponibilité dans la durée en régime stationnaire de 99% :

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

– Garantie de disponibilité et de temps de réponse (3 unités de temps) en régime stationnaire de 99% :

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

où req est une proposition atomique indiquant une réception de requête et acq est une proposition atomique indiquant une réponse à une requête. On notera qu'en réalité les deux occurrences de 99% n'ont pas la même signification. Celle correspondant à l'opérateur interne est une exigence sur le comportement du processus démarré en un état particulier tandis que la deuxième occurrence est une exigence globale sur les états pondérée par une distribution stationnaire. *A priori*, des valeurs différentes d'exigence auraient pu être spécifiées.

8.4.3. *Algorithme de vérification*

Etant données une CTMC et une formule ϕ de CSL, l'algorithme de vérification procède par évaluation successive des sous-formules de ϕ en «remontant» l'arbre syntaxique de la formule ϕ des feuilles à la racine et en étiquetant chaque état avec les sous-formules qu'il vérifie. Ainsi chaque étape de l'algorithme évalue une formule en interprétant les opérandes de l'opérateur le plus externe comme des propositions atomiques.

Nous sommes donc conduits à étudier chaque opérateur.

$\boxed{\phi = \neg\psi}$ L'algorithme étiquette avec ϕ chaque état non étiqueté avec ψ.

$\boxed{\phi = \psi \wedge \chi}$ L'algorithme étiquette avec ϕ chaque état étiqueté avec ψ et χ.

$\boxed{\phi = S_{\bowtie a}\psi}$ L'algorithme calcule la distribution stationnaire du processus démarré en s (ainsi qu'indiqué à la section 8.2.3). Puis il cumule les probabilités des états étiquetés par ψ et étiquette s avec ϕ si la quantité obtenue (disons p) vérifie $p \bowtie a$. Notons que pour les états d'une c.f.c. puits, un seul calcul est nécessaire pour tous les états de la c.f.c. De même, si la CTMC admet une unique distribution stationnaire alors la formule a une valeur de vérité indépendante de l'état.

$\boxed{\phi = P_{\bowtie a}\mathcal{X}^I\psi}$ Soit un état s, l'occurrence de la prochaine transition dans l'intervalle I et la satisfaction de ψ par l'état atteint sont deux événements indépendants. La probabilité recherchée est donc le produit de la probabilité de chacun de ces événements. Notons $I = [\tau, \tau']$; nous supposons ici sans perte de généralité que les intervalles sont fermés. En effet, en raison de la continuité des distributions, le fait que les bornes supérieures et inférieures de l'intervalle en fassent partie n'a pas d'incidence sur l'évaluation de la formule. Soit $\mathbf{Q}$ le générateur infinitésimal de la chaîne et $\mathbf{P}$ la matrice de transition de la chaîne incluse, alors la probabilité du premier événement est donnée par $e^{\tau\mathbf{Q}[s,s]} - e^{\tau'\mathbf{Q}[s,s]}$ tandis que celle du second événement est donnée par $\sum_{s' \vDash \psi} \mathbf{P}[s, s']$.

$\boxed{\phi = P_{\bowtie a}\psi\mathcal{U}^I\chi}$ L'évaluation de cette formule consiste essentiellement à effectuer une analyse transitoire de chaînes obtenues par des transformations élémentaires à partir de la chaîne originelle. Ainsi soit X une chaîne, alors X^ϕ est la chaîne obtenue en rendant absorbants les états qui vérifient ϕ. Afin de simplifier la présentation, nous étudions les différents types d'intervalle.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[0,\infty[}\chi$. Dans ce cas, la réalisation du processus doit rester dans des états qui vérifient ψ jusqu'à ce qu'un état vérifiant χ soit atteint et ceci sans contrainte de temps. Autrement dit, on suit le comportement de la chaîne jusqu'à ce qu'on rencontre un état qui vérifie $\neg\psi \vee \chi$. Etudions la chaîne $X^{\neg\psi\vee\chi}$. Si une c.f.c. puits de cette chaîne contient un état qui vérifie χ alors la probabilité recherchée est 1 pour tous les états de cette c.f.c. (car tous les états d'une c.f.c. puits sont récurrents) sinon

cette probabilité est nulle. Appelons une c.f.c. associée à une probabilité 1, une "bonne" c.f.c. Par conséquent, la probabibilité recherchée pour les états restants est égale à la probabilité d'atteindre un état d'une bonne c.f.c. Cette probabilité ne dépend que de la chaîne incluse de $X^{\neg\psi\vee\chi}$ et son calcul a déjà été décrit à la section 8.2.2.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[0,\tau]}\chi$. Dans ce cas, la réalisation du processus doit rester dans des états qui vérifient ψ jusqu'à ce qu'un état vérifiant χ soit atteint et ceci au plus tard à l'instant τ. Autrement dit on suit le comportement de la chaîne jusqu'à ce qu'on rencontre un état qui vérifie $\neg\psi \vee \chi$. La probabilité à calculer est donc égale à $\Pr(X^{\neg\psi\vee\chi}(\tau) \vDash \chi \mid X^{\neg\psi\vee\chi}(0) = s)$.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\tau]}\chi$. Dans ce cas, la réalisation du processus doit rester dans des états qui vérifient ψ durant l'intervalle $[0,\tau]$ et de plus vérifier χ à l'instant τ. On néglige la possibilité d'un changement d'état à l'instant τ car sa probabilité est nulle. Par conséquent, la probabilité à calculer est égale à $\Pr(X^{\neg\psi}(\tau) \vDash \psi \wedge \chi \mid X^{\neg\psi}(0) = s)$.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\infty[}\chi$. Dans ce cas, la réalisation du processus doit rester dans des états qui vérifient ψ durant l'intervalle $[0,\tau]$ puis à partir de l'état atteint s à l'instant τ verifier la formule $\psi\mathcal{U}^{[0,\infty[}\chi$. La probabilité recherchée est donc $\sum_{s'\vDash\psi}\Pr(X^{\neg\psi}(\tau) = s' \mid X^{\neg\psi}(0) = s)\cdot\boldsymbol{\pi}(s')$ où $\boldsymbol{\pi}(s')$ est calculée suivant la procédure du premier cas.

– $\phi = P_{\bowtie a}\psi\mathcal{U}^{[\tau,\tau']}\chi$. Un raisonnement similaire au cas précédent conduit la formule suivante pour la probabilité recherchée : $\sum_{s'\vDash\psi}\Pr(X^{\neg\psi}(\tau) = s' \mid X^{\neg\psi}(0) = s)\cdot\Pr(X^{\neg\psi\vee\chi}(\tau'-\tau) \vDash \chi \mid X^{\neg\psi\vee\chi}(0) = s')$.

8.5. Panorama de la vérification quantitative de chaînes de Markov

Historiquement, la vérification des chaînes à temps discret a précédé celle des chaînes à temps continu. La première approche de vérification de formules LTL sur des DTMC (proposée dans [VAR 85]) est conceptuellement simple : traduire la formule en un automate de Büchi, puis déterminiser cet automate en un automate de Rabin, effectuer le produit synchronisé de cet automate avec la DTMC ce qui conduit à une nouvelle DTMC sur laquelle une variante de l'analyse vue à la section 8.2 fournit la probabilité recherchée. Cependant la complexité de cet algorithme est doublement exponentielle relativement à la taille de la formule. Dans [COU 95], les auteurs construisent aussi une nouvelle DTMC en raffinant itérativement la DTMC initiale par une analyse des opérateurs de la formule. Ceci conduit à une procédure simplement exponentielle. Ils démontrent de plus qu'il s'agit là de la complexité optimale. C'est cette approche que nous avons suivie à la section 8.3.3. Un troisième algorithme [COU 03] traduit aussi la formule en un automate de Büchi. Cependant le choix de l'algorithme de traduction permet d'évaluer la probabilité associée à la formule directement à partir du produit synchronisé de l'automate et de la DTMC. Cette méthode a aussi une complexité théorique optimale et se comporte mieux dans les cas pratiques que la précédente.

Une technique classique d'analyse des modèles de performance consiste à associer des «récompenses» aux états et/ou aux transitions d'une chaîne et de calculer des indices de performance relatifs à ces récompenses. Afin d'étendre la portée de la vérification probabiliste à ce type de modèle, une nouvelle logique PRCTL est introduite dans [AND 03] accompagnée d'un algorithme d'évaluation de formules.

Les premiers travaux significatifs relatifs aux CTMC ont été établis dans [AZI 96, AZI 00]. Il y est démontré que la vérification de formules CSL sur les CTMC est décidable. Cependant l'algorithme correspondant est extrêmement complexe car il «s'interdit» les approximations que nous avons implicitement faites lors des calculs du paragraphe précédent.

De fait, même avec la méthode décrite plus haut, le calcul peut s'avérer impraticable pour des CTMC de grande taille. Une approche efficace pour faire face à ce problème consiste à tirer profit de la modularité de la spécification. On cherche alors à remplacer un module par un module plus petit mais équivalent vis à vis de la formule à vérifier. On procède ensuite à la vérifcation du modèle composé des modules réduits. Cette démarche initiée par [BAI 03a] a été généralisée dans [BAI 03b] où de nombreuses formes d'équivalence sont étudiées.

La logique CSL que nous avons introduite dans la section 8.4.2 présente deux limitations majeures. D'aune part, les formules de chemin ne portent que sur les états et ne peuvent exprimer qu'un changement d'état. D'autre part, les contraintes temporelles sont définies par un intervalle ce qui limite considérablement l'expression de contraintes de temps multiples le long d'un chemin. Le premier inconvénient a été résolu dans [BAI 04] car les opérateurs sont remplacés par une expression régulière portant à la fois sur les états et les actions conduisant à une logique appelée $asCSL$. Une approche radicalement différente est proposée dans [DON 07], puisque les formules sont exprimées à l'aide d'un automate temporisé déterministe à une horloge. Les auteurs démontrent que cette logique, notée CSL^{TA} étend strictement CSL et qu'elle est au moins aussi expressive que $asCSL$. De plus, son algorithme d'évaluation ne repose pas sur la construction de chaînes ad hoc mais sur la construction d'un processus de renouvellement markovien et l'étude de la chaîne discrete incluse associée à ce processus.

Une approche radicalement différente pour réduire la complexité de la vérification est proposée dans [YOU 06]. Supposons que nous devions vérifier la formule $P_{\leq a}\phi$, nous pouvons générer des exécutions aléatoires et calculer le ratio des exécutions qui vérifient ϕ ; en vertu de résultats classiques de probabilité, cette valeur tend vers la probabilité recherchée. Lorsque l'évaluation de la formule ϕ ne requiert qu'une exécution temporellement bornée, cette méthode est particulièrement efficace.

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).

[FOA 98] FOATA D., FUCHS A., *Calcul des probabilités*, Dunod, 1998, Seconde édition.

[FOA 02] FOATA D., FUCHS A., *Processus stochastiques. Processus de Poisson, chaînes de Markov et martingales*, Dunod, 2002.

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

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

[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.

[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.