Different Monotonicity Definitions in Stochastic Modelling*

Imène Kadi[1], Nihal Pekergin[2], and Jean-Marc Vincent[3]

[1] PRiSM, University Versailles-Saint-Quentin, 45 Av. des Etats-Unis 78000 France
[2] LACL, University Paris-Est, 61 avenue Général de Gaulle 94010, Créteil, France
[3] LIG, project MESCAL, 51, av. Jean Kuntzmann, 38330 Montbonnot, France
email : imene.kadi@prism.uvsq.fr
nihal.pekergin@univ-paris12.fr
jean-marc.vincent@imag.fr

Abstract. In this paper we discuss different monotonicity definitions applied in stochastic modelling. Obviously, the relationships between the monotonicity concepts depend on the relation order that we consider on the state space. In the case of total ordering, the stochastic monotonicity used to build bounding models and the realizable monotonicity used in perfect simulation are equivalent to each other while in the case of partial order there is only implication between them. Indeed, there are cases of partial order, where we can't move from the stochastic monotonicity to the realizable monotonicity, this is why we will try to find the conditions for which there are equivalences between these two notions. In this study, we will present some examples to give better intuition and explanation of these concepts.

1 Introduction

Simulation approaches constitute an alternative for performance evaluation, when numerical methods fail. In fact, they are usually used to model complex systems, such as, optical networks, distributed computer systems, stochastic Petri networks, and so on. In this paper we advocate the use of perfect simulation and combining this technique with stochastic monotonicity to speed up the computation. This method is based on the more general theory of coupling for Markov chains. Let us first review some ideas about coupling. Assume that we compute with the same random sequence of random numbers a sample path beginning at any initial state. If at time t two sample-paths are in the same state (we say that they couple), they will continue forever during all the simulation. When all the sample-paths have coupled, we obtain a sample state. We may use the state to initialize the simulation or consider it as a sample, thus it is not necessary anymore to continue the simulation.

It is known for a long time that coupling in the future does not provide samples distributed according to the steady-state distribution. But Propp and

* partially supported by french projects ANR-Blanc SMS, ANR- SETi06-02

Wilson have proved that coupling from the past (CFTP), also called backward-coupling, gives an exact sample of the steady-state distribution [11]. Coupling from the past is similar to coupling in the future but the initial time of the simulation will be chosen randomly whereas the final time is deterministic. In other words the Markov chain is not started at time 0 but sufficiently far away in the past such that at time 0 all the paths are coupled.

This method is extremely efficient. But many practical and theoretical problems remain to be solved for discrete Markovian systems to obtain a fully versatile technique. One of the problem we must consider is the number of operations we need to obtain a sample. The general backward algorithm tries to couple sample-paths beginning in every state in the state space. Thus modelling very large state space systems requires some model transformations. Furthermore the number of operations is at least linear in the size of the state space. The monotonicity property of the event structure of the model (which is formally defined in the next section) allows us to use a more efficient algorithm which sandwiches all sample-paths to couple into extreme ones.

We consider in this paper different monotonicity definitions applied in different context of stochastic modelling. First of them is the stochastic monotonicity concept associated to a stochastic ordering relation. This implies that the evolution of the underlying model is monotone regarding to the considered stochastic order. This monotonicity concept is one of the sufficient conditions to build bounding models [14]. For performability analysis of complex models, bounding models rather than the original one are considered to verify if performability requirements are satisfied by the original model. Obviously the bounding models must be easier to analyze than the original one [6].

In general the considered order relation on the state space is a total ordering. However the partial order is more suitable for multidimensional models. We explain first the stochastic monotonicity for a state space endowed with at least a pre-order and study the relationships with other monotonicity definitions.

The remaining monotonicity definitions are related to perfect simulation (sandwiching property). The first concept is called realizable monotonicity and was defined in [4]. The other definition is used in a software to provide perfect simulation of queueing networks (http://www-id.imag.fr/Logiciel/psi/). This is called event monotonicity and has been defined in more general terms in [7].

In this paper we present these definitions by emphasizing if the state space is totally ordered or not. We then compare them to give insights for the implications between them. We have considered relations between monotonicity definitions in a totally ordered state space [9]. In this case, the stochastic monotonicity and the event monotonicity are equivalent to each other. Therefore it is possible to construct bounding and stochastic monotone models in order to do monotone perfect simulations of systems which are not event monotone.

This paper is organized as follows: The next section is devoted to a brief presentation of considered stochastic models, perfect simulation, and stochastic ordering. In section 3, we give the different definitions of monotonicity: first the monotonicity in the sense of strong stochastic ordering then the realizable and event monotonicity used in perfect simulation. We present monotone perfect simulation of realizable monotone models in section 4. In section 5, we study the relationships between the stochastic monotonicity and the realizable monotonicity in order to see if stochastic monotone models can be used to perform monotone perfect simulation. So we show that these notions are different in the case of a totally and partially ordered state spaces. This is why we try to find cases of equivalence of these two notions under a partial order, and we give algorithms to construct event monotone systems in these cases.

2 Preliminaries

Markovian Discrete Event Systems (MDES) are dynamic systems evolving asynchronously and interacting at irregular instants called *event epochs* [7]. These systems are defined by means of a state space $\mathcal{X}$, a set of events $\mathcal{E}$, a set of probability measures $\mathcal{P}$, and a transition function Φ. $\mathbb{P}(e) \in \mathcal{P}$ denotes the occurrence probability of event $e \in \mathcal{E}$ while $\Phi(x, e)$ denotes the state to which the system moves from state x upon the occurrence of an event $e \in \mathcal{E}$.

Definition 1 (event) *An event e is an application defined on $\mathcal{X}$, that associates to each state $x \in \mathcal{X}$ a new state $y \in \mathcal{X}$.*

Definition 2 (transition function) *Let X_i be the state of the system at the i^{th} event occurrence time. The transition function $\Phi : \mathcal{X} \times \mathcal{E} \to \mathcal{X}$, defines the next state of the system X_{n+1} resulting from X_n upon the occurrence of an event e_{n+1}:*

$$X_{n+1} = \Phi(X_n, e_{n+1}) \tag{1}$$

Φ must obey to the following property to generate $\mathbf{P}$*:*

$$p_{ij} = \mathbb{P}(\phi(x_i, E) = x_j) = \sum_{e|\Phi(x_i,e)=x_j} \mathbb{P}(E = e) \tag{2}$$

Markov processes constitute a special, perhaps the most important subclass of stochastic processes [1]. We restrict ourselves here to the investigation of discrete state space and in that case refer to the stochastic processes as *chains*. Discrete Time Markov Chains(DTMC) are considered first, that is, Markov processes restricted to discrete, finite, or countably infinite state space, $\mathcal{X}$, and a discrete-parameter space T(time). For the sake of convenience, we set $T \subseteq \mathbb{N}_0$.

We consider in this work only time-homogeneous Markov chains, i.e, the conditional distribution function of a state X_{n+1} does not depend on observation time, that is, it is invariant with respect to time epochs n.

Definition 3 (DTMC) *A given stochastic process $\{X_0, X_1, ..., X_{n+1}, ...\}$ at the consecutive points of observation $0, 1, ..., n + 1$ constitutes a DTMC if the following relation on the conditional probability mass function(pmf), that is, the Markov property, holds for all $n \in \mathbb{N}$ and all $x_i \in \mathcal{X}$:*

$$\mathbb{P}(X_{n+1} = x_{n+1}|X_n = x_n, X_{n-1} = x_{n-1}, ..., X_0 = x_0) = \mathbb{P}(X_{n+1} = x_{n+1}|X_n = x_n). \tag{3}$$

Let $\mathcal{X} = \{0, 1, 2, ...\}$ *and write conveniently the notation for the conditional pmf of the process's one-step transition from state* i *to state* j *at time* n*:*

$$p_{ij}(n) = \mathbb{P}(X_{n+1} = x_{n+1} = j | X_n = x_n = i). \tag{4}$$

The one-step transition probability p_{ij} are given in a non-negative, stochastic [4] transition matrix $\mathbf{P}$:

$$\mathbf{P} = \mathbf{P}^{(1)} = [p_{ij}] \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ p_{20} & p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The following proposition gives how we can construct a transition function Φ for a time-homogeneous DTMC with a probability transition matrix $\mathbf{P}$ [8].

Definition 4 *A probability transition matrix* $\mathbf{P}$*, on a partially ordered state space* $(\mathcal{X}, \preceq)$*, can be described by a transition function* $\Phi : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$*, which defines the next state as*

$$X_{n+1} = \Phi(X_n, U_{n+1}) \tag{5}$$

where U *is a random variable taking values in an arbitrary probability space* $\mathcal{U}$*, such that, for all* $x, y \in \mathcal{X}$ *:* $\mathbb{P}(\Phi(x, U) = y) = p_{xy}$*.*

2.1 Perfect Sampling

Based on the transition function Φ, the following algorithm provides directly a sample of the steady state distribution. Let $\mathcal{X}$ be finite state space set.

Algorithm 1 Backward coupling simulation

n=1;
E[1]=Generate-event();
repeat
  n=2.n;
  for all $x \in \mathcal{X}$ **do**
    $Y[x] \leftarrow x$; {initialization of trajectories, size of vector Y is $|\mathcal{X}|$}
  end for
  for i=n downto n/2+1 **do**
    E[i]=Generate-event(); {generation of new events from -n/2 +1 to -n}
  end for
  for i=n downto 1 **do**
    $Y \leftarrow \Phi(Y, E[i])$; {generation of trajectories through events $E[i]$, }
  end for
  {$Y[x]$ is the state reached at time 0 for the trajectory issued from x at time -n}
until All $Y[x]$ are equal; {Coupling of all trajectories at time 0}

[4] The elements in each row of the matrix sum up to 1.

Let $\mathbb{E}\tau$ be the expectation of the coupling time, $|\mathcal{X}|$ be the size of the state space and $op(\Phi)$ be the average number of operations to compute the transited state. Clearly the average number of operations before coupling is $|\mathcal{X}|.\mathbb{E}\tau.op(\Phi)$.

Function Φ has a lot of influence on the number of operations. First the way it is implemented has a linear influence because of term $op(\Phi)$.

2.2 Stochastic ordering

Here we present the stochastic ordering of random variables and Markov chains. We refer to [14] for further informations. Let $\mathcal{X}$ be a discrete countable state space. We consider that $\mathcal{X}$ is endowed with at least a pre-order $\preceq$.The strong stochastic ordering associated to $\preceq$ will be denoted by $\preceq_{st}$.

A stochastic order can be defined by means of two approaches. The first way is to define them from a set of functions. The stochastic order defined in this case are called integral order. The second way is to define them from increasing sets which is more useful when the state space is not totally ordered.

Definition 5 (stochastic ordering) *Let X and Y be two random variables taking values on $\mathcal{X}$.*

$$X \preceq_{st} Y \Leftrightarrow \mathbb{E}f(X) \leq \mathbb{E}f(Y)$$

for all function $f : \mathcal{X} \to R$ which is not decreasing according to relation $\preceq$ whenever the expectations exist.

When the state space is totally ordered, the above definition implies the following property:

Property 1 *Let X and Y be two random variables taking values on $\mathcal{X}$, with a total order $\leqslant$, and let F_X and F_Y be respectively their distribution functions:*

$$X \leqslant_{st} Y \Leftrightarrow F_X(a) \geq F_Y(a), \forall a \in \mathcal{X}$$

From the order relation (at least pre-order) $\preceq$ on $\mathcal{X}$, we can define increasing sets on $\mathcal{X}$.

Definition 6 (increasing set) *Any subset Γ of $\mathcal{X}$ is called an increasing set if $x \preceq y$ and $x \in \Gamma$ implies $y \in \Gamma$.*

The stochastic order $\preceq_{st}$ is defined as follows from increasing sets:

Definition 7 (stochastic ordering with increasing sets) *Let T and V be two discrete random variables and Γ an increasing set defined on $\mathcal{X}$*

$$T \preceq_{st} V \Leftrightarrow \sum_{x \in \Gamma} \mathbb{P}(T = x) \leq \sum_{x \in \Gamma} \mathbb{P}(V = x), \ \ \forall \Gamma$$

3 Different definitions of monotonicity

Here we present different monotonicity definitions used in stochastic modelling. First we give the stochastic monotonicity associated to the stochastic order $\preceq_{st}$ then give the monotonicity definitions used for the perfect simulation.

3.1 Stochastic monotonicity

Following [14, 10] let us give the definition of the stochastic monotonicity for probability transition matrices of time-homogeneous DTMCs..

Definition 8 (stochastic monotonicity) *Let* $\mathbf{P}$ *be a stochastic matrix,* $\mathbf{P}$ *is st-monotone if and only if for any probability vectors on* $\mathcal{X}$*,* u *and* v*, if* $u \preceq_{st} v$ *implies that* $u\mathbf{P} \preceq_{st} v\mathbf{P}$.

Property 2 *Let* $\mathbf{P}$ *be the transition probability matrix of a time-homogeneous Markov chain* $\{X_n, n \geq 0\}$ *taking values in* $\mathcal{X}$ *endowed with relation order* $\preceq$. $\{X_n, n \geq 0\}$ *is st-monotone if and only if,*
$\forall (x, y) \mid x \preceq y$ *and* $\forall$ *increasing set* $\mathit{\Gamma} \in \mathcal{X}$

$$\sum_{z \in \mathit{\Gamma}} p_{xz} \leq \sum_{z \in \mathit{\Gamma}} p_{yz} \tag{6}$$

If the state space is totally ordered, the st-monotonicity implies that the rows of $\mathbf{P}$ are increasing:

Property 3 *In the case of totally ordered state spaces,* $\mathbf{P}$ *is st-monotone if and only if for all* i*, we have* $P_{i,*} \leqslant_{st} P_{i+1,*}$.

In the following example, we discuss the st-monotonicity by considering respectively a total order and then a partial order relation on the state space to show that there is no implication. Let us remark here that we consider partial orders compatible with the considered total order in the sense that the relations for the partial order exist also in the total order, but some states are not comparable under the partial order.

Example 1

$$\mathbf{P} = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 0 & 2/3 & 1/3 \end{pmatrix}$$

First we consider a total order: $\mathcal{X} = \{a, b, c, d\}$ *and* $a \leqslant b \leqslant c \leqslant d$*. We can see easily that the rows are increasing (property 3), so the matrix is stochastic monotone in the total ordering. Now we consider a partial order:* $a \preceq b \preceq d$*; and* $a \preceq c \preceq d$*. The increasing sets are* $\mathit{\Gamma}_1 = \{d\}$, $\mathit{\Gamma}_2 = \{c,\ d\}$, $\mathit{\Gamma}_3 = \{b, d\}$, $\mathit{\Gamma}_4 = \{b,\ c,\ d\}$, $\mathit{\Gamma}_5 = \{a,\ b,\ c,\ d\}$. $\mathbf{P}$ *is not monotone with respect to this order.*

For instance, for $\Gamma_3 = \{b, d\}$, the probability measure for row b is $1/6 + 1/3$, while this measure is $1/3$ for row d. Since $b \preceq d$, this violates the monotonicity.

Therefore we can see that the monotonicity with a total order does not imply the monotonicity with a partial order. From a first view, it may seem to be a contradiction, because with total order we must compare all of the rows, however with partial order we consider only comparable states. For example, we do not compare row b and c for partial order in this example. However we do not have the same increasing sets for these cases, for instance $\Gamma_3 = \{b, d\}$ is not an increasing set with total order.

Property 4 *If* $\mathbf{P}$ *is $\leqslant_{st}$-monotone with respect to a total order defined on $\mathcal{X}$, then* $\mathbf{P}$ *is not always $\preceq_{st}$-monotone with respect to a partial order defined on $\mathcal{X}$.*

3.2 Realizable monotonicity

First, we will give the definition of realizable monotonicity, used in Fill and Machida's works on the perfect simulation [5].

Definition 9 (realizable monotonicity) *Let* $\mathbf{P}$ *be a stochastic matrix defined on state space $\mathcal{X}$.* $\mathbf{P}$ *is said to be realizable monotone, if there exists a transition function Φ as in Eq. 5, such that Φ preserves the order relation i.e. for all $u \in \mathbf{U}$, we have $\Phi(x, u) \preceq \Phi(y, u)$, whenever $x \preceq y$.*

There is an other definition of monotonicity used to perform perfect simulation of finite queuing networks by software Psi2 [15].

Definition 10 (event monotonicity) *The underlying model is said to be event monotone, if the transition function by events (Eq. 1) preserves the order ie. for each $e \in \mathcal{E}$*

$$\forall (x, y) \in \mathcal{X} \;\; x \preceq y \Longrightarrow \Phi(x, e) \preceq \Phi(y, e)$$

This notion of event monotonicity is the same as the realizable monotonicity if the set of events $\mathcal{E}$ is pre-defined. So a system is realizable monotone means that there exists a finite set of events $\mathcal{E}$ for which the system is event monotone. In the case of finite DTMCs, the cardinality of the set of events is upper bounded by the number of non null entries of the transition matrix.

Example 2 *Let $(\mathcal{X}, \preceq)$ be a partial ordering state space, $\mathcal{X} = \{a, b, c, d\}$, $a \preceq b \preceq d$ and $a \preceq c \preceq d$;*

We consider three events with the following probabilities $p_{e_1} = 1/6$, $p_{e_2} = 1/3$, $p_{e_3} = 1/2$.

$$\mathbf{P} = \begin{pmatrix} 1/2 & 1/3 & 0 & 1/6 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 1/3 & 0 & 1/6 \\ 0 & 1/3 & 1/6 & 1/2 \end{pmatrix}$$

	1/6	1/6	1/6	1/6	1/6	1/6
a	a			b		d
b	a			b	d	
c	a			b		d
d	b		c	d		

	e_3	e_2	e_1
a	a	b	d
b	a	d	b
c	a	b	d
d	b	d	c

If we consider the initial set of event, we can see from the first table that $\mathbf{P}$ *is realizable monotone, but it is not event monotone, for instance we have in the second table, for event* $p_{e_1} = 1/6$ *,* $\Phi(b, e_1) = b$ *is incomparable with* $\Phi(d, e_1) = c$*. But if we change the set of events, and define new events following the first table, we obtain an event monotone system. For instance, we can, from the table of realizable monotonicity, divide the interval [0,1] into monotone events, we obtain five events with the following probabilities* $p_{e_1} = 1/3$*,* $p_{e_2} = 1/6$*,* $p_{e_3} = 1/6$*,* $p_{e_4} = 1/6$*,* $p_{e_4} = 1/6$*.*

	p_{e_1}	p_{e_2}	p_{e_3}	p_{e_4}	p_{e_5}
a	a	a	b	b	d
b	a	a	b	d	d
c	a	a	b	b	d
d	b	c	d	d	d

We summarize the relationships between these types of monotonicity by the following scheme 1. We can see that there is no implication between monotonicity under the total order and a partial order compatible with the total order neither for the stochastic monotonicity nor the realizable monotonicity. When the state space is totally ordered, both monotonicity notions are equivalent while for partially ordered state spaces the realizable monotonicity implies the stochastic monotonicity.

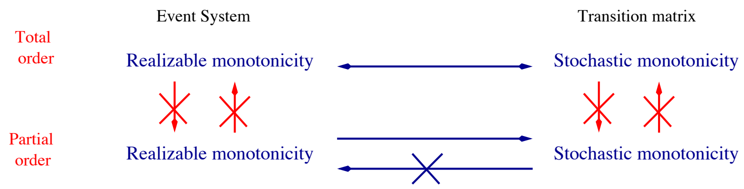

Fig. 1. Relations between monotonicity notions in total and partial order

4 Realizable monotonicity and perfect sampling

When the operator Φ is realizable monotone, the algorithm could be simplified by making iteration only on maximal and minimal values of the state space. If the trajectories issued from minimal and maximal states are coupled, due to the realizable monotonicity, trajectories issued from all other states are also coupled. The perfect simulation of monotone models will clearly reduce the computation and memory complexity to obtain a sample [15].

We give in the following backward-coupling for event monotone models. Let us turn now to the expectation of the coupling time for event-monotone systems. In the algorithm M (resp. m) denotes the set of maximal (resp. minimal) elements in the state space. This algorithm has the same convergence properties as Algorithm (1). Thus the expected number of operations is $(M+m).\mathbb{E}\tau_1.op(\Phi)$.

Algorithm 2 Backward-coupling simulation (event monotone version)

n=1;
E[1]=Generate-event();
repeat
    n=2.n;
    for all $x \in \mathcal{M} \cup m$ **do**
        $Y[x] \leftarrow x$; {initialization of trajectories,size of vector Y is $|M \cup m|$}
    end for
    for i=t downto t/2+1 **do**
        E[i]=Generate-event(); {generation of new events from -n/2 +1 to -n}
    end for
    for i=n downto 1 **do**
        $Y \leftarrow \Phi(Y, E[i])$; {generation of trajectories through events $E[i]$, }
    end for{$Y[x]$ is the state reached at time 0 for the trajectory issued from x at time -n}
until All $Y([x]$ are equal; {Coupling of maximal and minimal trajectories at time 0}

5 Stochastic monotonicity and perfect simulation

Now we discuss how one can perform a monotone perfect simulation of a stochastic monotone DTMC. So we will study the relations between the stochastic monotonicity and the realizable monotonicity, and find the conditions that allow us to move from a stochastic monotone DTMC to an event monotone MDES.

5.1 Totally ordered state space

When the state space is totally ordered the stochastic monotonicity and the realizable monotonicity are equivalent [9]. However the stochastic monotonicity is necessary but not sufficient for realizable monotonicity for partially ordered state spaces. [4]

Theorem 1 *When the state space is totally ordered ($\leqslant$), the stochastic monotonicity and the realizable monotonicity are equivalent.*

This result has already been proved, but for a better comprehension we will give a proof to this theorem.

Proof. – Realizable monotonicity $\Longrightarrow$ Stochastic monotonicity From the realizable monotonicity definition, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x \ \leqslant\ y\ then\ \forall u \in [0,1] : \Phi(x,u) \leqslant \Phi(y,u) \tag{7}$$

The function Φ is the inverse probability distribution function. Let X and Y be two random variables corresponding respectively to rows x and y of $\mathbf{P}$. So $\Phi(x,u) = F_X^{-1}$, $\forall u$.
From equation 7, we obtain :

$$F_X^{-1}(u) \leqslant F_Y^{-1}(u),\ \forall u$$

this implies that for each state $a \in \mathcal{X}$:

$$F_X(a) \geq F_Y(a)$$

It follows from the definition of the strong stochastic ordering (property 1) that $X \leqslant_{st} Y$. Thus, the model is stochastically monotone.

– Stochastic monotonicity $\Longrightarrow$ Realizable monotonicity
From the stochastic monotonicity, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x\ \leqslant\ y\ then\ P[x,*] \preceq_{st} P[y,*] \tag{8}$$

Let X and Y be two random variables corresponding respectively to rows x and y of $\mathbf{P}$. From equation 8 and property 1 of strong stochastic ordering we obtain:

$$F_X(a) \geq F_Y(a),\ \forall a \in \mathcal{X} \tag{9}$$

Let u be a random variable, uniformally distributed in [0,1]. The equation 9 implies that :

$$\forall u \in [0,1] : F_X^{-1}(u) \leqslant F_Y^{-1}(u)$$

We see that the function F^{-1} satisfies the conditions of monotonicity. So we can always find a monotone transition function for the system.

Example 3 *Let* $\mathbf{P}3$ *be a transition matrix defined on a total ordered state space*$(\mathcal{X}, \leqslant)$*,* $\mathcal{X} = \{a,b,c,d\}$ *and* $a \leqslant b \leqslant c \leqslant d$*.*

$$\mathbf{P}3 = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 1/6 & 1/2 & 1/3 \end{pmatrix}$$

It can be easily verified that $\mathbf{P}3$ *is st-monotone. This model can be described by a transition function* Φ*, obtained by the inverse probability distribution function.*

<table>
<tr><th></th><th>1/6</th><th>1/6</th><th>1/6</th><th>1/6</th><th>1/6</th><th>1/6</th></tr>
<tr><td>a</td><td colspan="3">a</td><td>b</td><td colspan="2">c</td></tr>
<tr><td>b</td><td colspan="3">a</td><td>b</td><td colspan="2">d</td></tr>
<tr><td>c</td><td colspan="3">a</td><td>c</td><td colspan="2">d</td></tr>
<tr><td>d</td><td colspan="3">b</td><td>c</td><td colspan="2">d</td></tr>
</table>

It can be easily seen from the table that this model is realizable monotone.

5.2 Partially ordered state spaces

We now consider a partial order on the state space and show that there is only implication but not the equivalence between these two monotonicity definitions.

Theorem 2 *In the case of partially ordered state spaces, if the system is realizable monotone, it is also stochastically monotone.*

Proof. By means of equation 2 and property 2, we can rewrite stochastic monotonicity constraints of matrix $\mathbf{P}$ as follows

$$\forall(x,y)|x \preceq y \textit{ and } \forall \Gamma, \sum_{z\in\Gamma}\sum_{u|\phi(x,u)=z} \mathbb{P}(U=u) \leq \sum_{z\in\Gamma}\sum_{u|\phi(y,u)=z} \mathbb{P}(U=u)$$

From the realizable monotone definition, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x\ \preceq\ y\ then\ \forall u \in [0,1] : x' = \Phi(x,u) \preceq \Phi(y,u) = y'$$

Thus if x' belongs to an increasing set Γ, then y' belongs to this set (definition (6). The above inequalities are satisfied for all increasing set Γ, thus $\mathbf{P}$ is st-monotone.

The reciprocal of this implication is not true. We will prove it by a counter example: We consider a transition matrix $\mathbf{P3}$ in a partially ordered state space. $\mathcal{X} = \{a,b,c,d\}$ and $a \preceq b \preceq d$; $a \preceq c \preceq d$.

$$\mathbf{P3} = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \end{pmatrix}$$

It can be easily verified that $\mathbf{P3}$ is st-monotone. This model can be described by transition function Φ, obtained by the inverse probability distribution function by considering the total order $a \leqslant b \leqslant c \leqslant d$.

<table>
<tr><td></td><td>1/6</td><td>1/6</td><td>1/6</td><td>1/6</td><td>1/6</td><td>1/6</td></tr>
<tr><td>a</td><td colspan="3">a</td><td>b</td><td colspan="2">c</td></tr>
<tr><td>b</td><td colspan="2">a</td><td colspan="2">b</td><td colspan="2">d</td></tr>
<tr><td>c</td><td colspan="3">a</td><td>c</td><td colspan="2">d</td></tr>
<tr><td>d</td><td colspan="2">b</td><td colspan="2">c</td><td colspan="2">d</td></tr>
</table>

It can be seen from the table that it is not realizable monotone, for instance, we have for $u \in [3/6, 4/6]$ $\Phi(a,u) = b$ is incomparable with $\Phi(c,u) = c$.

We can not find another transition function which makes this system realizable monotone.

Proposition 1 *It is not possible to construct a realizable monotone transition function for the above example.*

Proof. Since $b \preceq d$ and $c \preceq d$, the transitions from states $b,\ c,\ d$ to state d with probability $1/3$ must be associated to the same interval u. Similarly, since $a \preceq b$ and $a \preceq c$ the transitions from states $a,\ c$ to state a with probability $1/2$ must be associated to the same interval u, the transitions from states $a,\ b$ to state a with probability $1/2$ must be associated to the same interval u .

So, for states b, and c it remains only an interval of $u_e = 1/3$ to assign. For b the transition which is not associated is to state b, and for c there are two transitions, one is to state a and the other is to state c. Now, we discuss the case of state a, where $a \preceq c$ and $a \preceq b$. For state a, we have an interval of $1/2$ to assign, the transitions which are not associated are to state b and c. If we associate b to the interval u_e, we have a case of non comparability with state $\Phi(c, u_e) = c$. Similarly, if we associate c to the interval u_e, we will have a case of non comparability with state $\Phi(b, u_e) = b$. Thus it is not possible to build a realizable monotone transition function.

5.3 Case of equivalence in partial order

We will give a case of partial order for which there is an equivalence between the stochastic monotonicity and the realizable monotonicity, we will then give an algorithm to construct the monotone transition function Φ which can be used in Psi2 to do monotone perfect simulation.

Theorem 3 *When the state space is partially ordered in a tree, if the system is stochastic monotone, then there exists a finite set of events* $e_1, e_2, ..., e_n$, *for which the system is event-monotone.*

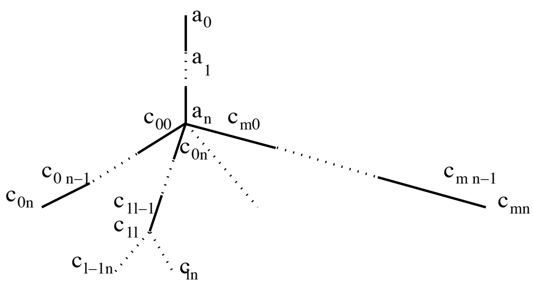

Fig. 2. State space which is partially ordered as a tree

We consider one strongly connected component. Let $A = \{a_1 \leq a_2 \leq ...a_n\}$ be the states which are comparable with all others. This set contains at least the root of the tree. $F = \{f_1, ..., f_m\}$ denotes the set of leaves. Suppose that there are m branches from a_n to each leaf f_i. The branches from a_n to f_i are called $C_i = \{c_{i0}, ..., c_{in} = f_i\}$, where c_{i0} is the successor of a_n. Obviously, the states in a branch are totally ordered. We consider branch by branch. and for a given branch C_i, we determine for all states $x \in \mathcal{X}$, events e_h, such that $\Phi(x, e_h) = c_{ij}$. Let N be the number of states in $\mathcal{X}$.

Now we will give the algorithm that construct the monotone transition function Φ, the idea of this algorithm can be summarized as follows: we consider branch by branch and for branch C_i we find events which trigger transition to a state of C_i. Then we consider the states of A and find events which trigger transition to a state of A.

Algorithm 3 Stochastic monotonicity $\rightarrow$ event-monotonicity

```
E = ∅{the set of events is initially empty}
for k ∈ {1, ··· m} do
   {Consider branch C_k}
   V = [v_1, v_2, ..., v_N]{a vector representing the column index of the rightmost posi-
   tive values for each row}
   repeat
      for i ∈ {1, 2 ··· N} do
         for j ∈ {v_i, ··· , c_{k,l}, c_{k,l−1}, ··· c_{k,0}, a_n, ··· a_0} do
            if p_{i,j} = 0 then
               j ← j + 1
            end if
         end for
         v_i ← j {update vector V}
      end for
      h ← h + 1 {the next event e_h}
      p_{e_h} ← min_{1≤i≤N} p_{i,v_i}{probability for event e_h}
      for i = 1 to N do
         Φ(i, e_h) ← v_i
         p_{i,v_i} ← p_{i,v_i} − p_{e_h}{update matrix P}
      end for
   until Σ_{e_h∈E} p_{e_h} = max_{x∈F}(Σ_{i=0}^{n} p_{x,c_{ki}})
end for
repeat
   for i = 1 to N do
      j ← v_i
      while p_{i,j} = 0 do
         j ← j − 1
      end while
      v_i ← j {update vector V}
   end for
   h ← h + 1 {the next event e_h}
   p_{e_h} ← min_{1≤i≤N} p_{i,v_i}{probability for event e_h}
   for i = 1 to N do
      Φ(i, e_h) ← v_i
      p_{i,v_i} ← p_{i,v_i} − p_{e_h}{update matrix P}
   end for
until Σ_{i=1}^{h} p_{e_h} = 1
```

Proof. To prove this algorithm, we must show that for all two comparable states x and y, if $x \preceq y$, than we can find a transition function Φ, such that $\Phi(x,u) \preceq \Phi(y,u)$, $\forall u$. For each branch k of the tree, we have from the stochastic monotonicity the following proprieties

$$p_{xc_{kn}} \le p_{yc_{kn}}$$

$$p_{xc_{kn-1}} + p_{xc_{kn}} \le p_{yc_{kn-1}} + p_{yc_{kn}}$$

. . .

$$p_{xc_{k0}} + ... + p_{xc_{kn-1}} + p_{xc_{kn}} \le p_{yc_{k0}} + ... + p_{yc_{kn-1}} + p_{yc_{kn}}$$

These proprieties satisfy the same conditions of the stochastic monotonicity in a total order. This means that for each branch of the tree, we can construct a monotone transition function by the same method used in the total order. Now, if $\sum_{i=0}^{n} p_{yc_{ki}} > \sum_{i=0}^{n} p_{xc_{ki}}$ we must prove that for all u in the interval, which represent $\sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}$, $\Phi(y,u) \succeq \Phi(x,u)$.

Let

$$diff_k = \sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}$$

So, we must show that the sum of all the differences $diff_k$ is smaller than $(\sum p_{xa} + p_{xb})$. This can be verified in the following equation:

$$\sum p_{xa} + p_{xb} = \sum_{k=0}^{m} diff_k + \sum p_{ya} + p_{yb} \tag{10}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} = \sum_{k=0}^{m} (\sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}) + \sum p_{ya} + p_{yb}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} + \sum_{k=0}^{m} (\sum_{i=0}^{n} p_{xc_{ki}}) = \sum_{k=0}^{m} (\sum_{i=0}^{n} p_{yc_{ki}}) + \sum p_{ya} + p_{yb}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} + \sum_{k=0}^{m} (\sum_{i=0}^{n} p_{xc_{ki}}) = 1 = \sum_{k=0}^{m} (\sum_{i=0}^{n} p_{yc_{ki}}) + \sum p_{ya} + p_{yb}$$

This last equation is evident because of the stochastic proprieties of the matrix.

6 Conclusion

In this paper, we study different monotonicity notions used in stochastic modelling. The stochastic monotonicity associated to stochastic ordering relation and the event and realizable monotonicity is used in perfect simulation. The monotonicity concept depends on the relation order that we consider on the state space. First, we show that if we have a monotone model on a total order, this does not imply that it is monotone in the partial order for both monotonicity notions.

Additionally, we have discussed the relationships between the stochastic monotonicity and the monotonicity used to perform perfect simulation, in order to see whether it is feasible to do monotone perfect simulation on a stochastic monotone models. There are different mathematical tools to build bounding models for complex discrete event systems. In conclusion, under a total order, the different monotonicity definitions are equivalent to each other. However, under a partial order, the realizable monotonicity implies the stochastic monotonicity. In fact, we have shown that stochastic monotonicity are not sufficient to obtain an event monotone model, but we must verify others conditions on the DTMC. For instance if the partial order is a tree, we have proved that there is an equivalence between the two notions of monotonicity, and we have developed an algorithm which construct the realizable monotone transition function Φ, to do perfect monotone simulation.

References

1. G. Bolch, S. Greiner, H. de Meer, and K. Trivedi. *Queueing Networks and Markov Chains*. John Wiley & Sons, 1998.
2. A.A. Borovkov and S. Foss. Two ergodicity criteria for stochastically recursive sequences. *Acta Appl. Math.*, 34, 1994.
3. P. Diaconis and D. Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, 1999.
4. James Allen FILL and Motoya Machida. An interruptible algorithm for perfect sampling via markov chains. Department of Mathematical sciences, The Johns Hopkins university.
5. James Allen FILL and Motoya Machida. Realizable monotonicity and inverse probability transform, department of mathematical sciences, the johns hopkins university.
6. J.M. Fourneau and N. Pekergin. An algorithmic approach to stochastic bounds. In *LNCS 2459, Performance evaluation of complex systems: Techniques and Tools*, pages 64–88, 2002.
7. P. Glasserman and D. Yao. *Monotone Structure in Discrete-Event Systems*. John Wiley & Sons, 1994.
8. Olle Haggstrom. *Finite Markov Chains and algorithmic applications*. 2001. Matematisk Statistik, Chalmers teknisk hogshola och Goteborgs universitet.
9. N. Pekergin J. Vienne J.M. Fourneau, I. Kadi and J.M. Vincent. Perfect simulation and monotone stochastic bounds. *Valuetools. Nantes , France*, 2007.
10. William MASSEY. Stochastic ordering for markov processes on partially ordered spaces. May 1987, Department of Mathematical sciences, The Johns Hopkins university.
11. J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1&2):223–252, 1996.
12. O Stenflo. *Ergodic theorems fory Iterated Function Systems controlled by stochastic sequences*. Doctoral thesis n. 14, Umea university, 1998.
13. O Stenflo. Ergodic theorems for markov chains represented by iterated function systems. *Bull. Polish Acad. Sci. Math*, 2001.
14. Dietrich Stoyan. Comparison methods for queues and other stochastic models. 1983.
15. Jean-Marc Vincent. Perfect simulation of queueing networks with blocking and rejection. Laboratoire ID-IMAG, Projet INRIA APACHE.