Monotonicity requirements for efficient exact sampling with Markov chains *

PAWEŁ LOREK
University of Wrocław

PIOTR MARKOWSKI
University of Wrocław

Created: 16.11.2015
Last modified: June 22, 2017

Abstract

We recall three methods for exact sampling from a stationary distribution of a Markov chain: the coupling from the past (CFTP) algorithm, a method based on strong stationary duality (SSD), and Fill's rejection algorithm. Each method, to be applied efficiently, requires a different notion of monotonicity, which is defined with respect to a partial ordering of the state space, namely realizable monotonicity, Möbius monotonicity, and stochastic monotonicity. We show full relations between monotonicities. The applicability of the CFTP algorithm implies the applicability of Fill's rejection algorithm, but does not imply that of the SSD-based method. We also state one open problem related to these monotonicities.

Keywords: Exact simulation, perfect simulation, coupling from the past, Fill's rejection algorithm, strong stationary duality, strong stationary time, coupling, stochastic monotonicity, realizable monotonicity, Möbius monotonicity, failure rate monotonicity, Siegmund duality

Mathematics Subject Classification (2010) 60J10 • 60G40 • 60J80

1 Introduction

Monte Carlo Markov Chain (MCMC) methods are a class of tools for approximate sampling from a given distribution (usually intractable by other methods). The method is based on constructing an ergodic Markov chain that has the desired distribution as its stationary distribution. Then the algorithm outputs a state after simulating the chain for some number of steps. The ergodicity implies that the more steps are performed, the closer it is to the stationary distribution. However, to say something about the error, one needs to have some theoretical bounds on the rate of convergence, e.g., the mixing time. In many practical problems this is an obstacle that is hard to overcome.

Exact (or perfect) simulation refers to the art of converting a Markov chain (usually obtained from MCMC methods) into an algorithm which returns an unbiased sample from its stationary distribution. In this paper we briefly present three such algorithms. Our main focus in this paper is on their monotonicity requirements for *efficient* application. Each of the algorithms requires then a different notion of monotonicity. The monotonicities will be defined with respect to a partial ordering (in the applications, a state space usually has some natural underlying partial ordering). The idea of all the algorithms is based on a coupling.

Coupling from the past (CFTP) is probably the most famous exact sampling algorithm, introduced in a ground breaking paper [PW96]. The ingenious idea of the algorithm is to realize the chain as a stochastic flow and evolve it *from the past* (rather than into the future). Doing so requires considering coupled realizations of chains started at all possible starting points. This is infeasible in most cases. However, if the chain is so-called **realizable monotone**, we need only to

*Work supported by NCN Research Grant DEC-2013/10/E/ST1/00359

simulate two chains, thus making the algorithm very effective. Although many variations of the algorithm have been invented, often with slightly different requirement for monotonicity, see, e.g., [HN98], [Hub03], [Ken98], [KM00], we focus on this (widely used) monotonicity.

The second exact sampling algorithm is *Strong Stationary Dual*-based. The notion was introduced in [DF90b] and exploited mainly for studying the rate of convergence. However, having such a dual chain (including a so-called *link*), we can couple two chains in such a way that when the dual chain hits a specific state, then the original chain has the stationary distribution. The point is that there is no general way to come up with such a dual chain. In [DF90b], the authors give a recipe only for chains whose time reversals are stochastically monotone. This duality for total ordering has been exploited in many contexts, see, e.g., [DSC06], [Fil09b], [Fil09a], [FL14] (most of them deal only with birth and death chains). However, many interesting distributions are given on a state space possessing a natural non-linear ordering. The existence (and recipe) of such a dual for a partial ordering was given in [LS12]. We briefly present an exact sampling algorithm based on the coupling from [DF90b] and the dual given in [LS12]. This dual exists (and thus we can use the algorithm) if and only if the time reversal is **Möbius monotone**. For examples of Möbius monotone chains, see [LS16], for connections with Siegmund duality on partially ordered state spaces, see [Lor] and [Lor17].

The last algorithm we will present is the so-called **Fill's rejection algorithm** [Fil98]. The author presents the algorithm already assuming some monotonicity. We present here a more general version of the algorithm, together with a short proof of correctness. Similarly to the previous algorithms, it is hard to apply it to a general chain. However, the algorithm can be applied assuming **stochastic monotonicity** (w.r.t. the partial ordering) of the time reversed chain. For more details and/or extensions of the algorithm, consult, e.g., [FMMR00], [Dim01], [CLR01].

The three exact sampling algorithms presented, require, as already mentioned, three different monotonicities for efficient application: realizable monotonicity, Möbius monotonicity, and stochastic monotonicity. The relation between the first and the last one is already known: for a general partial ordering, realizable monotonicity implies stochastic monotonicity, whereas they are equivalent for total or tree-orderings, see [FM01], [Mac01]. However, Möbius monotonicity has not been studied as extensively as the other two. Theorem 5.1 and Fig. 1 show all the relationships between the orderings (including also weak monotonicities and distinguishing between Möbius-$^{\uparrow}$ and Möbius-$^{\downarrow}$ monotonicities). In particular, one interesting (from both points of view: theoretical and practical) implication is that the chain (and/or its time reversal) does not have to be stochastically monotone (and thus realizable monotone) but can be Möbius monotone. Thus we can still use one of the exact sampling algorithms. We also present several examples of orderings and chains showing all possible cases of being/not being monotone in a specific sense. However we are forced to leave one **open problem** (see open problem 1): we cannot prove or disprove that there exists a chain and ordering such that the chain is Möbius-$^{\uparrow}$, Möbius-$^{\downarrow}$, stochastically monotone, but is not realizable monotone.

The organization of this paper is as follows: In Section 2 we present the aforementioned exact sampling algorithms in their full generality. In Section 3 we formally introduce the monotonicities. Section 4 contains applications of the monotonicities the three algorithms mentioned above (i.e., to their efficient applications). In Section 5 we present the relationship between the monotonicities, whereas the examples are postponed to the Appendix.

2 Three general methods for exact sampling

Throughout the paper we consider an ergodic Markov chain $\mathbf{X} = \{X_k\}_{k\geqslant 0}$ with the initial distribution ν, finite state space $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$, transition matrix $\mathbf{P} = [\mathbf{P}(\mathbf{e}, \mathbf{e}')]_{\mathbf{e},\mathbf{e}'\in\mathbb{E}}$, and stationary distribution π. The distribution of the chain at step k, started with the initial distribution ν, is denoted by $\nu\mathbf{P}^k(\cdot)$. For a measure f on $\mathbb{E}$ we write $f(A) = \sum_{\mathbf{e}\in A} f(\mathbf{e})$ for $A \subseteq \mathbb{E}$. It is said that $\overleftarrow{\mathbf{X}}$ is the time reversed chain of the chain $\mathbf{X}$ if $\overleftarrow{\mathbf{X}}$ is defined on the same space as $\mathbf{X}$ and has t.m. $\overleftarrow{\mathbf{P}}(\mathbf{e}, \mathbf{e}') = \frac{\pi(\mathbf{e}')}{\pi(\mathbf{e})}\mathbf{P}(\mathbf{e}', \mathbf{e})$.

One can simulate the Markov chain using an update rule:

Definition 2.1. *A function* $\phi : \mathbb{E} \times [0,1] \to \mathbb{E}$ *is an* ***update rule*** *for the chain* $\mathbf{X}$ *with the transition matrix* $\mathbf{P}$ *if:*

1. *for fixed* $\mathbf{e} \in \mathbb{E}$ *the function* $\phi(\mathbf{e}, u)$ *is piecewise constant, and*
2. *for all* $\mathbf{e}, \mathbf{e}' \in \mathbb{E}$ *we have* $\int_0^1 \mathbb{1}(\phi(\mathbf{e}, u) = \mathbf{e}')du = \mathbf{P}(\mathbf{e}, \mathbf{e}')$.

Note that for the uniformly distributed random variable $U \sim Unif[0,1]$ we have $\mathbf{P}(\mathbf{e}, \mathbf{e}') = Pr(\phi(\mathbf{e}, U) = \mathbf{e}')$. Having such an update rule, one can recursively simulate the chain:

$$X_0 \sim \nu, \quad X_{k+1} = \phi(X_k, U_{k+1}), \tag{1}$$

where $U_1, U_2, \ldots$ is an iid sequence of random variables uniformly distributed on $[0,1]$.

The usual Monte Carlo Markov Chain (MCMC) provides methods for approximate sampling from the desired distribution π, usually intractable by other methods (such as, e.g., inverting the distribution function). Roughly speaking, the methods include constructing an ergodic chain with π being its stationary distribution. Thus, simulating $X_0, X_1, \ldots$ *long enough*, the distribution of X_k will be close to the stationary distribution (since ergodicity implies that $\lim_{k\to\infty} \nu\mathbf{P}^k(\cdot) = \pi(\cdot)$). Note that: i) usually it will never be exactly the stationary distribution; ii) to know how close it is to the stationarity the distribution of X_k, one needs to know the rate of convergence.

We briefly recall three methods for **exact sampling** (often called *perfect sampling*), i.e., obtaining an unbiased sample from π. All of them rely on the concept of *coupling*. A coupling of a pair of Markov chains with a common transition matrix $\mathbf{P}$ is a bivariate process $\{(X_k, Y_k)\}_{k\geqslant 0}$ such that marginally $\{X_k\}_{k\geqslant 0}$ and $\{Y_k\}_{k\geqslant 0}$ are Markov chains with the transition matrix $\mathbf{P}$ (in particular, the processes may be, and usually are, dependent, and have different initial distributions).

2.1 Method 1: Coupling from the past

One of the most known algorithms for exact sampling is called coupling from the past (CFTP) (or the Propp–Wilson algorithm, cf. [PW96]). Given an increasing sequence $N_1, N_2, \ldots$ of positive integers (usually $N_r = 2^{r-1}$), the algorithm is as follows:

Algorithm 1 Coupling from the past (CFTP).

Require: State space $\mathbb{E}$, ergodic chain $\mathbf{X}$ with update rule ϕ.

1: Set $n = 1$.
2: For each $\mathbf{e} \in \mathbb{E}$ simulate the Markov chain starting at time $-N_n$ in state $\mathbf{e}$ and run it till time 0 using the same update rule ϕ and iid random variables $U_{-N_n+1}, U_{-N_n+2}, \ldots, U_{-1}, U_0$ uniformly distributed on $[0,1]$ (the same for each chain).
3: If all chains in the previous step end up in the same state $\mathbf{e}_0$ at time 0, then output $\mathbf{e}_0$ and **stop**.
4: Set $n = n + 1$ and go to Step 2 (keep the previously used $\{U_i\}_{0\leqslant i\leqslant -N_n+1}$ for new n).

The very rough idea of the CFTP algorithm is the following: assume at some time "in the past," say at $-N_n$, for each $\mathbf{e} \in \mathbb{E}$ we started a chain. Later on, using the same update rule and the same uniform random variables driving the chains, all the chains have coalesced before time 0. If we had started the chains earlier, even at "minus infinity," but from $-N_n$ on using the same uniform random variables, we would still end up in the same state at time 0. And from "minus infinity" till 0 it surely has already "reached" stationarity, thus the output of the algorithm is a random variable with the distribution being the stationary distribution of the chain. For more technical details (e.g., that it always terminates), see [PW96].

2.2 Method 2: Fill's rejection algorithm

Let $p(\cdot), q(\cdot)$ be two probability distributions on $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$. Assume that for some $c \in \mathbb{R}$ we have $p(\mathbf{e}) \leqslant cq(\mathbf{e})$ for all $\mathbf{e} \in \mathbb{E}$ (then $c \geqslant 1$, but $c > 1$ if the distributions are different). The

classical acceptance-rejection algorithm lets us simulate from $p(\cdot)$ if we are able to simulate from $q(\cdot)$:

Algorithm 2 Acceptence-rejection algorithm.

Require: Distributions $p(\cdot)$, $q(\cdot)$ on $\mathbb{E}$, constant $c : p(\mathbf{e}) \leqslant cq(\mathbf{e})$.

1: Generate $Y \sim q(\cdot)$.
2: Flip a coin with Head probability $\frac{p(Y)}{cq(Y)}$. If Head, then return $X := Y$.
3: Else go to Step 1.

Based on this algorithm, Fill [Fil98] came up with a tricky idea for simulating from a stationary distribution π of an ergodic Markov chain. We will present here its slightly generalized version (without specific assumptions on the transition matrix). We include also a short proof of its correctness (although it is similar to that of Fill). Fix an integer $k \geqslant 1$ (time instance) and a state $\mathbf{e}_1 \in \mathbb{E}$. In the above settings, we want to simulate from $p(\cdot) = \pi(\cdot)$, given that we are able to simulate from $q(\cdot) = P(X_k = \cdot | X_0 = \mathbf{e}_1) = \mathbf{P}^k(\mathbf{e}_1, \cdot)$ (which can be done straightforwardly). Let $\overleftarrow{\phi}$ be an update function of the time reversed chain $\overleftarrow{\mathbf{X}}$ with t.m. $\overleftarrow{\mathbf{P}}(\mathbf{e}, \mathbf{e}') = \frac{\pi(\mathbf{e}')}{\pi(\mathbf{e})}\mathbf{P}(\mathbf{e}', \mathbf{e})$. Similarly as in the CFTP algorithm, we can use a sequence $N_r = 2^{r-1}$.

Algorithm 3 Fill's rejection algorithm.

Require: State space for all the following chains $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$, ergodic chain $\mathbf{X}$ with a transition matrix $\mathbf{P}$.

1: Set $n = 1$.
2: Simulate the Markov chain $\mathbf{X}$ starting at time 0 in state $\mathbf{e}_1$ and run it till time $k = N_n$ using the transition matrix $\mathbf{P}$. Denote $X_k = \mathbf{e}_z$.
3: Treat $(X_k = \mathbf{e}_z, X_{k-1}, \ldots, X_0 = \mathbf{e}_1)$ as a path of the time reversed chain $(\overleftarrow{X}_0, \ldots, \overleftarrow{X}_k)$. For $s = 0, \ldots, n$ do: Assume $\overleftarrow{X}_s = \mathbf{e}$ and $\overleftarrow{X}_{s+1} = \mathbf{e}'$. Then generate $U_s \sim \texttt{Unif}\{u : \overleftarrow{\phi}(\mathbf{e}, u) = \mathbf{e}'\}$.
4: Start M chains $\overleftarrow{\mathbf{Y}}^j, j = 1, \ldots, M$ so that $\overleftarrow{Y}^j_0 = j$ and *couple* them simulating $\overleftarrow{Y}^j_{s+1} = \phi(\overleftarrow{Y}^j_s, U_{s+1})$ (using the common update function ϕ and the randomness obtained in the previous step).
5: If all chains $\overleftarrow{\mathbf{Y}}^j, j = 1, \ldots, M$ have coupled before time k (and thus at time k they are all in state $\mathbf{e}_1$), output $\mathbf{e}_z$ and **stop**.
6: Erase all information, set $n = n + 1$ and go to Step 2.

In Step 3 we simulate random variables $U_1, \ldots, U_k$ in such a way that if we started the time reversed chain at $\mathbf{e}_z$ then we would obtain exactly the trajectory $(X_k = \mathbf{e}_z, X_{k-1}, \ldots, X_0 = \mathbf{e}_1)$. And this is what will happen for sure with one of $\overleftarrow{\mathbf{Y}}^j$ (the one starting in $\mathbf{e}_z$). That is why if all $\overleftarrow{\mathbf{Y}}^j, j = 1, \ldots, M$ coalesce, then we must have $\overleftarrow{Y}^j_k = \mathbf{e}_1$ for all $j = 1, \ldots, M$. Let $C_k(\mathbf{e})$ denote the event that all the chains $\overleftarrow{\mathbf{Y}}^j$ have coalesced before time k and that at this time they are all in $\mathbf{e}$. Of course we have for any $\mathbf{e}$ that $\overleftarrow{\mathbf{P}}^k(\mathbf{e}, \mathbf{e}_z) \leqslant \Pr[C_k(\mathbf{e}_z)]$. Now we are ready to choose the constant c from Alg. 2.

$$\frac{\pi(\mathbf{e})}{\mathbf{P}^k(\mathbf{e}_1, \mathbf{e})} = \frac{\pi(\mathbf{e}_1)}{\overleftarrow{\mathbf{P}}^k(\mathbf{e}, \mathbf{e}_1)} \leqslant \frac{\pi(\mathbf{e}_1)}{\Pr[C_k(\mathbf{e}_1)]} =: c.$$

Thus we simulate from distribution $\mathbf{P}^k(\mathbf{e}_1, \cdot)$, say $\mathbf{e}_z$ was obtained. We should accept $\mathbf{e}_z$ with probability

$$\frac{\pi(\mathbf{e}_z)}{c\mathbf{P}^k(\mathbf{e}_1, \mathbf{e}_z)} = \frac{\Pr[C_k(\mathbf{e}_1)]}{\pi(\mathbf{e}_1)} \frac{\pi(\mathbf{e}_z)}{\mathbf{P}^k(\mathbf{e}_1, \mathbf{e}_z)} = \frac{\Pr[C_k(\mathbf{e}_1)]}{\overleftarrow{\mathbf{P}}^k(\mathbf{e}_z, \mathbf{e}_1)}.$$

The whole point is that this is exactly the acceptance probability in Alg. 3:

$$\Pr(\overleftarrow{Y}^1_k = \ldots = \overleftarrow{Y}^M_k = \mathbf{e}_1 \mid \overleftarrow{X}_0 = \mathbf{e}_z, \overleftarrow{X}_k = \mathbf{e}_1, \overleftarrow{Y}^j_0 = \mathbf{e}_j, j = 1, \ldots, M)$$

$$= \frac{\Pr(\overleftarrow{Y}_k^1 = \ldots = \overleftarrow{Y}_k^M = \mathbf{e}_1 \mid \overleftarrow{X}_0 = \mathbf{e}_z,\ \overleftarrow{Y}_0^j = \mathbf{e}_j, j = 1, \ldots, M)}{\Pr(\overleftarrow{X}_k = \mathbf{e}_1 \mid \overleftarrow{X}_0 = \mathbf{e}_z,\ \overleftarrow{Y}_0^j = \mathbf{e}_j, j = 1, \ldots, M)}.$$

In this setting, X_0 and $\mathbf{Y}^j$ are independent, thus

$$= \frac{\Pr(\overleftarrow{Y}_k^1 = \ldots = \overleftarrow{Y}_k^M = \mathbf{e}_1 \mid \overleftarrow{Y}_0^j = \mathbf{e}_j, j = 1, \ldots, M)}{\Pr(\overleftarrow{X}_t = \mathbf{e}_1 \mid \overleftarrow{X}_0 = \mathbf{e}_z)} = \frac{\Pr[C_k(\mathbf{e}_z)]}{\overleftarrow{\mathbf{P}}^k(\mathbf{e}_z, \mathbf{e}_1)}.$$

2.3 Method 3: Strong stationary duality

A random variable T is a **Strong Stationary Time** (SST) for $\mathbf{X}$ if it is a stopping time independent from X_T such that X_T has the distribution π. It was introduced in [AD86] mainly for studying the rate of convergence of the chain, but it is also applicable for exact sampling, simply by simulating the chain until time T we obtain unbiased sample from π. Although there are many examples where such SST was find (probably the best example is Top-To-Random card shuffling), the problem is that the examples were usually found "ad hoc", in general it is not easy to come up with SST.

Diaconis and Fill [DF90b] came up with a systematic way of finding an SST, which we will describe here directly with application to exact sampling.

Let $\mathbb{E}^* = \{\mathbf{e}_1^*, \ldots, \mathbf{e}_N^*\}$ be the state space of an absorbing Markov chain $\mathbf{X}^*$ with initial distribution ν^* and transition matrix $\mathbf{P}^*$, whose unique absorbing state is denoted by $\mathbf{e}_N^*$. An $N \times M$ matrix Λ is said to be a *link* if it is a stochastic matrix such that $\Lambda(\mathbf{e}_N^*, \mathbf{e}) = \pi(\mathbf{e})$ for all $\mathbf{e} \in \mathbb{E}$. We say that $\mathbf{X}^*$ is a **strong stationary dual** (SSD) of $\mathbf{X}$ with link Λ if

$$\nu = \nu^* \Lambda \quad \text{and} \quad \Lambda \mathbf{P} = \mathbf{P}^* \Lambda. \tag{2}$$

In this paper we assume that the SSD has the same state space, i.e., $\mathbb{E}^* = \mathbb{E}$. For the general case, see [DF90b]. The sample path of the chain $\mathbf{X}^*$ can be constructed from a sample path of $\mathbf{X}$ as follows. Start with $X_0 = \mathbf{e}^0$ and (using additional randomness) set

$$X_0^* = \mathbf{e}^{*0} \quad \text{with probability} \quad \frac{\nu^*(\mathbf{e}^{*0})\Lambda(\mathbf{e}^{*0}, \mathbf{e}^0)}{\nu(\mathbf{e}^0)}.$$

Then we proceed as follows. Assume $X_0 = \mathbf{e}^0, \ldots, X_{k-1} = \mathbf{e}^{(k-1)}$ and $X_0^* = \mathbf{e}^{*0}, \ldots, X_{k-1}^* = \mathbf{e}^{*(k-1)}$. If $X_k = \mathbf{e}^k$ have been chosen, then set

$$X_k^* = \mathbf{e}^{*k} \text{ with probability } \frac{\mathbf{P}^*(\mathbf{e}^{*(k-1)}, \mathbf{e}^{*(k)})\Lambda(\mathbf{e}^{*k}, \mathbf{e}^k)}{\triangle(\mathbf{e}^{*(k-1)}, \mathbf{e}^k)},$$

where $\triangle = \mathbf{P}^*\Lambda$. This construction yields a bivariate chain (X_k^*, X_k) such that $\Pr(X_k = \cdot | X_0^* = \mathbf{e}^{*0}, \ldots, X_k^* = \mathbf{e}^{*k}) = \Lambda(\mathbf{e}^{*k}, \cdot)$ (consult [DF90a], [DF90b]). This implies that T, the first time the chain $\mathbf{X}^*$ hits the state $\mathbf{e}_N^*$ (the absorbing one) and the value of X_T are independent. Moreover, the distribution of X_T is $\Lambda(\mathbf{e}_N^*, \cdot) = \pi(\mathbf{e})$.

In summary, we are able to couple two chains in such a way that when one hits a specific state $(\mathbf{e}_N^*)$ then the other has a stationary distribution. This way we can obtain an unbiased sample from π, i.e., we can perform an exact sampling. Note that having the SST T lets one also study the rate of convergence (the main application of this duality in [DF90b]): the time to absorption T for $\mathbf{X}^*$ is an SST for $\mathbf{X}$. In many examples, SST have been found ad hoc. The above duality approach provided the first systematic way of finding them. Below we present the above mentioned description of SSD-based exact sampling in algorithmic form.

Algorithm 4 Exact sampling based on SSD.

Require: Ergodic chain $\mathbf{X}$ and absorbing chain $\mathbf{X}^*$ on the same state space $\mathbb{E}$, link Λ.

1: Start with $X_0 = \mathbf{e}^0$ and set $X_0^* = \mathbf{e}^{*0}$ with probability $\frac{\nu^*(\mathbf{e}^{*0})\Lambda(\mathbf{e}^{*0},\mathbf{e}^0)}{\nu(\mathbf{e}^0)}$.
2: If $X_0^* = \mathbf{e}_N^*$ then output X_0 and **stop**.
3: Set $n = 1$.
4: Having $X_{n-1} = \mathbf{e}^{n-1}$ set $X_n = \mathbf{e}^n$ with probability $P(e^{n-1}, e^n)$.
5: Having $X_{n-1} = \mathbf{e}^{n-1}, X_n = \mathbf{e}^n, X_{n-1}^* = \mathbf{e}^{*n-1}$ set $X_n^* = \mathbf{e}^{*n}$ with probability $\frac{\mathbf{P}^*(\mathbf{e}^{*(n-1)},\mathbf{e}^{*(n)})\Lambda(\mathbf{e}^{*n},\mathbf{e}^n)}{\triangle(\mathbf{e}^{*(n-1)},\mathbf{e}^n)}$, where $\triangle = \mathbf{P}^*\Lambda$.
6: If $X_n^* = \mathbf{e}_N^*$ then output X_n and **stop**.
7: Set $n = n + 1$ and go to Step 4 (keep previously simulated $X_n = \mathbf{e}^n, X_n^* = \mathbf{e}^{*n}$ for new n).

3 Monotonicites in Markov chains

In the previous section we briefly described some methods for exact sampling. Note however that CFTP and Fill's rejection algorithm, as they stand, are very inefficient (the number of chains one has to simulate is equal to the cardinality of $\mathbb{E}$) and no concrete way for finding the SSD was given (how to choose/find Λ and $\mathbf{P}^*$). This is where monotonicities come into play. Each of the methods can be **efficiently** applied if the chain is monotone in some way.
So far we did not need any structure on $\mathbb{E}$. However in many examples there is a natural ordering of the state space, e.g., a total ordering, a coordinatewise ordering, etc. From now on we assume that $\mathbb{E}$ is equipped with a partial ordering $\preceq$, making $(\mathbb{E}, \preceq)$ a poset. We also assume that $\mathbf{e}_1$ is the minimum and $\mathbf{e}_M$ is the maximum. We will use the following notation. We say that $U \in \mathbb{E}$ is an **upset** if $(\mathbf{e}_1 \preceq \mathbf{e}_2, \mathbf{e}_1 \in U) \Rightarrow \mathbf{e}_2 \in U$. Similarly, we say that $D \in \mathbb{E}$ is a **downset** if $(\mathbf{e}_1 \preceq \mathbf{e}_2, \mathbf{e}_2 \in D) \Rightarrow \mathbf{e}_1 \in D$. For given $\mathbf{e} \in \mathbb{E}$ we define $\{\mathbf{e}\}^\uparrow := \{\mathbf{e}' : \mathbf{e} \preceq \mathbf{e}'\}$ and $\{\mathbf{e}\}^\downarrow := \{\mathbf{e}' : \mathbf{e}' \preceq \mathbf{e}\}$. Note that each $\{\mathbf{e}\}^\uparrow$ ($\{\mathbf{e}\}^\downarrow$) is an upset (downset), but, in general, not vice versa.

All the monotonicities we are about to define are defined for chains on a common state space $\mathbb{E}$ with respect to a fixed partial ordering $\preceq$. By $\mathbf{X} \in \mathcal{P}$ we mean that $\mathbf{X}$ has monotonicity property $\mathcal{P}$ keeping in mind that it is defined w.r.t. the fixed partial ordering $\preceq$.

Usual and weak stochastic monotonicity.

Definition 3.1. *A Markov chain $\mathbf{X}$ with transition matrix $\mathbf{P}$ is **stochastically monotone** (we write $\mathbf{X} \in \mathcal{S}$) if and only if for all upsets U and all $\mathbf{e} \preceq \mathbf{e}' \in \mathbb{E}$ we have $\mathbf{P}(\mathbf{e}, U) \leqslant \mathbf{P}(\mathbf{e}', U)$.*

Remark 3.2. Since the complement of any upset is a downset, the condition for stochastic monotonicity can be equivalently given by: for all downsets D and all $\mathbf{e} \preceq \mathbf{e}' \in \mathbb{E}$ we have $\mathbf{P}(\mathbf{e}, D) \geqslant \mathbf{P}(\mathbf{e}', D)$.

Stochastic monotonicity can be equivalently defined in the following way. For two random variables Y_1, Y_2 (with distribution functions ν_1, ν_2) on $\mathbb{E}$, we say that $Y_1 \preceq_{st} Y_2$ (or $\nu_1 \preceq_{st} \nu_2$) $\iff$ $E[f(Y_1)] \leqslant E[f(Y_2)]$ for all nondecreasing (w.r.t. $\preceq$) functions $f : \mathbb{E} \to \mathbb{R}$. Then the Markov chain $\mathbf{X}$ with the transition matrix $\mathbf{P}$ is stochastically monotone if and only if $\nu_1 \preceq_{st} \nu_2$ implies $\nu_1\mathbf{P} \preceq_{st} \nu_2\mathbf{P}$.
Recall that K is an *upward kernel* if it is a Markov kernel such that $K(\mathbf{e}_i, \cdot)$ is supported on $\{\mathbf{e}_j \in \mathbb{E} : \mathbf{e}_i \preceq \mathbf{e}_j\}$. The following lemma goes back to Strassen [Str65] (and is part of Theorem 1 in [KKO77]).

Lemma 3.3. *A markov chain $\mathbf{X}$ with transition matrix $\mathbf{P}$ is stochastically monotone if and only if for all $\mathbf{e} \preceq \mathbf{e}'$ there exists an upward kernel $K_{\mathbf{e},\mathbf{e}'}$ such that $\mathbf{P}(\mathbf{e}', \mathbf{e}_j) = \sum_{\mathbf{e}_i : \mathbf{e}_j \preceq \mathbf{e}_i} \mathbf{P}(\mathbf{e}, \mathbf{e}_i) K_{\mathbf{e},\mathbf{e}'}(\mathbf{e}_i, \mathbf{e}_j)$.*

Replacing any upset (downset) in Definition 3.1 with a specific one we obtain the notion of weak monotonicity.

Definition 3.4. *A Markov chain* $\mathbf{X}$ *with transition matrix* $\mathbf{P}$ *is* ***weakly-$\uparrow$ monotone*** *(we write* $\mathbf{X} \in \mathcal{W}^{\uparrow}$*) if and only if for all* $\mathbf{e} \preceq \mathbf{e}', \mathbf{e}_j \in \mathbb{E}$ *we have* $\mathbf{P}(\mathbf{e}, \{\mathbf{e}_j\}^{\uparrow}) \leqslant \mathbf{P}(\mathbf{e}', \{\mathbf{e}_j\}^{\uparrow})$.
The chain is ***weakly-$\downarrow$ monotone*** *(we write* $\mathbf{X} \in \mathcal{W}^{\downarrow}$*) if and only if for all* $\mathbf{e} \preceq \mathbf{e}', \mathbf{e}_j \in \mathbb{E}$ *we have* $\mathbf{P}(\mathbf{e}, \{\mathbf{e}_j\}^{\downarrow}) \geqslant \mathbf{P}(\mathbf{e}', \{\mathbf{e}_j\}^{\downarrow})$.

We define $\mathcal{W} := \mathcal{W}^{\uparrow} \cap \mathcal{W}^{\downarrow}$.

Realizable monotonicity. This notion of monotonicity is defined in terms of the update rule of the chain given in Definition 2.1.

Definition 3.5. *A Markov chain* $\mathbf{X}$ *with transition matrix* $\mathbf{P}$ *is* ***realizable monotone*** *if there exists a monotone update rule (preserving the ordering), i.e.,*

$$\forall (u \in [0,1])\ \forall (\mathbf{e} \preceq \mathbf{e}') \quad \phi(\mathbf{e}, u) \preceq \phi(\mathbf{e}', u).$$

This definition implies that for any states $\mathbf{e} \preceq \mathbf{e}'$ and upset U we have

$$\phi(\mathbf{e}, u) \in U \Rightarrow \phi(\mathbf{e}', u) \in U. \tag{3}$$

Finding a monotone update rule is often a challenging task, and proving that none exist can be even harder.

Möbius monotonicity.
We can identify the ordering $\preceq$ with the matrix $\mathbf{C}(\mathbf{e}_i, \mathbf{e}_j) = \mathbb{1}(\mathbf{e}_i \preceq \mathbf{e}_j)$. We can always rearrange the states in such a way that $\mathbf{C}$ is upper triangular (keeping in mind that the enumerations of the states in $\mathbf{C}$ and $\mathbf{P}$ must preserve the same order), thus invertible. The inverse of $\mathbf{C}$ is usually denoted by $\mu \equiv \mathbf{C}^{-1}$ and called the **Möbius function**.

Definition 3.6. *The function* $\mathbf{f} : \mathbb{E} \to \mathbb{R}^M$ *is* ***Möbius-$\downarrow$*** *(****Möbius-$\uparrow$****) monotone if* $\mathbf{f}(\mathbf{C}^{\mathsf{T}})^{-1} \geqslant 0$ *(*$\mathbf{f}\mathbf{C}^{-1} \geqslant 0$*), i.e., each entry is nonnegative.*

Definition 3.7. *A Markov chain* $\mathbf{X}$ *with transition matrix* $\mathbf{P}$ *is* ***Möbius-$\downarrow$ monotone*** *(we write* $\mathbf{X} \in \mathcal{M}^{\downarrow}$*) if*

$$\mathbf{C}^{-1}\mathbf{P}\mathbf{C} \geqslant 0 \quad \textit{(each entry nonnegative)}.$$

Equivalently, in terms of the transition probabilities,

$$\forall (\mathbf{e}_i, \mathbf{e}_j \in \mathbb{E}) \qquad \sum_{\mathbf{e} \succeq \mathbf{e}_i} \mu(\mathbf{e}_i, \mathbf{e}) \mathbf{P}(\mathbf{e}, \{\mathbf{e}_j\}^{\downarrow}) \geqslant 0.$$

The chain is ***Möbius-$\uparrow$ monotone*** *(we write* $\mathbf{X} \in \mathcal{M}^{\uparrow}$*) if*

$$(\mathbf{C}^{\mathsf{T}})^{-1}\mathbf{P}\mathbf{C}^{\mathsf{T}} \geqslant 0 \quad \textit{(each entry nonnegative)}.$$

In terms of the transition probabilities this is

$$\forall (\mathbf{e}_i, \mathbf{e}_j \in \mathbb{E}) \qquad \sum_{\mathbf{e} \preceq \mathbf{e}_i} \mu(\mathbf{e}, \mathbf{e}_i) \mathbf{P}(\mathbf{e}, \{\mathbf{e}_j\}^{\uparrow}) \geqslant 0.$$

We define $\mathcal{M} := \mathcal{M}^{\uparrow} \cap \mathcal{M}^{\downarrow}$.
In the applications, checking Möbius monotonicity is usually not harder than checking stochastic monotonicity. First, note that the inverse of $\mathbf{C}$ (i.e., the Möbius function of the ordering) is known for many natural partial orderings (however, its derivation is often not trivial). To mention a few:

- **E1** For $\mathbb{E} = \{1, \ldots, M\}$ and a linear ordering $\preceq := \leqslant$ the Möbius function is given by $\mu(i,i) = 1$, $\mu(i, i+1) = -1$ and $\mu(i,j) = 0$ for $j \notin \{i, i+1\}$.
- **E2** For $\mathbb{E} = \{0,1\}^d$ with the coordinate-wise partial ordering $\mathbf{e} \preceq \mathbf{e}'$, if $\mathbf{e}(i) \leqslant \mathbf{e}'(i), i = 1, \ldots, d$, the Möbius function is given by $\mu(\mathbf{e}, \mathbf{e}') = (-1)^{|\mathbf{e}| - |\mathbf{e}'|}$ if $\mathbf{e} \preceq \mathbf{e}'$ and 0 otherwise (where $|\mathbf{e}| = \sum_{i=1}^{d} \mathbf{e}(i)$).

E3 For $\mathbb{E} = \{0, 1, \ldots, N\}^d$ with coordinate-wise partial ordering $\mathbf{e} \preceq \mathbf{e}'$, if $\mathbf{e}(i) \leqslant \mathbf{e}'(i), i = 1, \ldots, d$, the Möbius function is given by $\mu(\mathbf{e}, \mathbf{e}') = (-1)^{|\mathbf{e}'|-|\mathbf{e}|}$ if $\mathbf{e}'(k) = \mathbf{e}(k)$ or $\mathbf{e}'(k) = \mathbf{e}(k) + 1$ for each $k = 1, \ldots, d$.

E4 For a finite set I let $P(I)$ be the set of all partitions of I. Let $\alpha, \beta \in P(I)$. The typically considered partial order is the following: $\alpha \preceq \beta$ if $\forall(A \in \alpha)\exists(B \in \beta)(A \subseteq B)$. As derived in [Com70], the Möbius function is given by $\mu(\alpha, \beta) = \mathbf{1}_{\alpha \preceq \beta}(-1)^{|\alpha|+|\beta|} \prod_{B \in \beta}(l_B^\alpha - 1)!$, where l_B^α is the number of atoms from α in $B \in \beta$.

Checking the Möbius monotonicity of a chain having a Möbius function turns out to be feasible in many cases. For the total ordering (**E1**), exemplary calculations are given in [Lor]. The computations checking Möbius-$^\downarrow$ monotonicity for some nonsymmetric random walk on the cube (**E3**) are given in [LS12]. For the chain corresponding to a nonstandard queue network, the computations are given in [Lor]. The partial ordering on partitions (**E4**) was considered in the context of duality in [HM16].

4 Applications of monotonicites

4.1 Realizable monotonicity and an efficient coupling from the past algorithm

The CFTP algorithm given in Alg. 1 is very inefficient. The number of chains we have to run is equal to the size of the state space. In most cases where CFTP is to be applied, the size of the state space is huge (e.g., exponential in some parameter). The main idea of the algorithm was to run the chains "from the past" and check if all of them have coupled before time 0. Note that if we have a monotone update rule and say $X_{-m} = \mathbf{e} \preceq X'_{-m} = \mathbf{e}'$, then

$$X_{-m+1} = \phi(X_{-m}, U_{-m+1}) \preceq \phi(X'_{-m}, U_{-m+1}) = X'_{-m+1}.$$

Thus it is enough to start only two chains: $X_0^1 = \mathbf{e}_1$ and $X_0^2 = \mathbf{e}_M$. Summarizing, if the chain is *realizable monotone* and has the minimum and the maximum, then we have an efficient CFTP algorithm:

Algorithm 5 Efficient coupling from the past.

Require: State space $\mathbb{E}$, ergodic chain $\mathbf{X}$, monotone update rule ϕ

1: Set $n = 1$
2: Start two chains at time $-N_n$, one at the minimum $\mathbf{e}_1$, the other at the maximum $\mathbf{e}_M$. Run the chains till time 0 using the same update rule ϕ and iid random variables $U_{-N_n+1}, U_{-N_n+2}, \ldots, U_{-1}, U_0$ uniformly distributed on $[0, 1]$ (the same for each chain).
3: If both chains in previous step end up in the same state $\mathbf{e}_0$ at time 0, then output $\mathbf{e}_0$ and **stop**.
4: Set $n = n + 1$ and go to Step 2 (keep previously used $\{U_i\}_{0 \leqslant i \leqslant -N_n+1}$ for new n).

4.2 Stochastic monotonicity and an efficient Fill's rejection algorithm

Similarly to the general CFTP algorithm given in Alg. 1, Fill's rejection Alg. 3 is very inefficient. This is due to the fact that we have to start (and simulate) as many chains as there are elements of the state space. It turns out that the algorithm can be made efficient by assuming stochastic monotonicity of the time reversed chain $\overleftarrow{X}$. (This condition is weaker, as forthcoming sections will show, than being realizable monotone).

Assume for the moment that $\overleftarrow{X}$ is *realizable monotone* (this will soon be relaxed to stochastic monotonicity). Assume that $\mathbf{e}_i \preceq \mathbf{e}_j$ and that $\overleftarrow{Y}_s^i = \mathbf{e}_i \preceq \overleftarrow{Y}_s^j = \mathbf{e}_j$ for some $s \leqslant k$. Realizable

monotonicity implies that $\overleftarrow{Y}^i_{s+1} = \phi(\overleftarrow{Y}^i_s, U_s) \preceq \phi(\overleftarrow{Y}^j_s, U_s) = \overleftarrow{Y}^j_{s+1}$. This means that then in Step 5 of the algorithm, checking the coalescence of all M chains is equivalent to checking only that the chain $\overleftarrow{\mathbf{Y}}^M$ (the one started in $\mathbf{e}_M$) has already reached the minimum $\mathbf{e}_1$ at time k. In other words, it is enough to simulate just one chain $\overleftarrow{\mathbf{Y}}^M$.
Now we relax the realizable monotonicity requirement, assuming only that $\overleftarrow{X}$ is stochastically monotone. Similarly, we want to have an efficient version of the algorithm simulating only one chain $\overleftarrow{\mathbf{Y}}^M$ (denoted simply by $\mathbf{Y}$). This time we do not generate U_s as in Step 3 of Alg. 3. We make use of Lemma 3.3 instead. Assume that at some step s we have $\overleftarrow{X}_s = \mathbf{e}_1$, $\overleftarrow{X}_{s+1} = \mathbf{e}_2$ and $\overleftarrow{Y}_s = \mathbf{e}_i$. Then, since $\mathbf{e}_1 \preceq \mathbf{e}_2$, we may choose a state $\mathbf{e}_j$ for $\overleftarrow{Y}_{s+1}$ with probability $K_{\mathbf{e}_1,\mathbf{e}_i}(\mathbf{e}_2, \mathbf{e}_j)$. This constructions ensures that $\overleftarrow{X}_s \preceq \overleftarrow{Y}_s, s = 0, \ldots, k$. Thus, similarly, only the condition $\overleftarrow{Y}_k = \mathbf{e}_1$ must be checked. In summary, we have:

Algorithm 6 Efficient Fill's rejection algorithm.

Require: State space $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$, ergodic chain $\mathbf{X}$ whose time reversal $\overleftarrow{X}$ is stochastically monotone and set of kernels $K_{\mathbf{e},\mathbf{e}'}$ for all $\mathbf{e} \preceq \mathbf{e}'$.

1: Set $n = 1$
2: Simulate the Markov chain $\mathbf{X}$ starting at time 0 in state $\mathbf{e}_1$ and run it till time $k = N_n$ using the transition matrix $\mathbf{P}$. Denote $X_t = \mathbf{e}_z$
3: Simulate the chain $\overleftarrow{\mathbf{Y}}$ starting at the maximum, i.e., $\overleftarrow{Y}_0 = \mathbf{e}_M$ in the following way: Assume at time s we have $\overleftarrow{X}_s = \mathbf{e}_1$, $\overleftarrow{Y}_s = \mathbf{e}_i$ and $\overleftarrow{X}_{s+1} = \mathbf{e}_2$. Set $\overleftarrow{Y}_{s+1} = \mathbf{e}_j$ with probability $K_{\mathbf{e}_1,\mathbf{e}_i}(\mathbf{e}_2, \mathbf{e}_j)$.
4: If $\overleftarrow{Y}_k = \mathbf{e}_1$, then output $\mathbf{e}_z$ and **stop**
5: Erase all information, set $n = n + 1$ and go to Step 2.

4.3 Möbius monotonicity and strong stationary duality

In Section 2.3 we presented an exact sampling algorithm based on strong stationary duality. Note however that no recipe was given on how to find such a dual. The duality was introduced in [DF90b], where the recipe was given only in case the time reversed chain $\overleftarrow{\mathbf{X}}$ was stochastically monotone w.r.t. a total ordering. In [LS12] an extension to partial orderings was given. Surprisingly, it turned out that not the usual stochastic monotonicity, but rather Möbius monotonicity, was required. We recall here the main theorem from [LS12].

Theorem 4.1 (Lorek and Szekli [LS12])**.** *Let $\mathbf{X}$ be an ergodic Markov chain on a finite state space $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$ which is partially ordered by $\preceq$ and has the maximum $\mathbf{e}_M$. For a stationary distribution π and an initial distribution ν we assume that*

(i) $g(\mathbf{e}) = \frac{\nu(\mathbf{e})}{\pi(\mathbf{e})}$ is Möbius-$\downarrow$ monotone,

(ii) $\overleftarrow{\mathbf{X}}$ is Möbius-$\downarrow$ monotone.

Then there exists a strong stationary dual chain $\mathbf{X}^*$ on $\mathbb{E}^* = \mathbb{E}$ with link a truncated stationary distribution $\Lambda(\mathbf{e}_j, \mathbf{e}_i) = \mathbb{1}(\mathbf{e}_i \preceq \mathbf{e}_j)\frac{\pi(\mathbf{e}_i)}{H(\mathbf{e}_j)}$, where $H(\mathbf{e}_j) = \sum_{\mathbf{e}:\mathbf{e}\preceq\mathbf{e}_j} \pi(\mathbf{e})$. The initial distribution and the transitions of $\mathbf{X}^*$ are given, respectively, by

$$\nu^*(\mathbf{e}_i) = H(\mathbf{e}_i) \sum_{\mathbf{e}:\mathbf{e}\succeq\mathbf{e}_i} \mu(\mathbf{e}_i, \mathbf{e}) g(\mathbf{e}), \tag{4}$$

$$\mathbf{P}^*(\mathbf{e}_i, \mathbf{e}_j) = \frac{H(\mathbf{e}_j)}{H(\mathbf{e}_i)} \sum_{\mathbf{e}:\mathbf{e}\succeq\mathbf{e}_j} \mu(\mathbf{e}_j, \mathbf{e}) \overleftarrow{\mathbf{P}}(\mathbf{e}, \{\mathbf{e}_i\}^\downarrow). \tag{5}$$

(The Möbius monotonicity of the function $g(\mathbf{e})$ means that the resulting $\nu^*(\mathbf{e})$ is nonnegative).

Remark 4.2. Note that the existence of the minimum is not required in Theorem 4.1. However, if it exists and if the chain $\mathbf{X}$ starts at the minimum (i.e., $\Pr(X_0 = \mathbf{e}_1) = 1$), then so does the dual chain (i.e., $\Pr(X_0^* = \mathbf{e}_1)$). Similarly one can construct an SSD chain when there is a minimum $\mathbf{e}_1$ and the time reversed chain $\overleftarrow{\mathbf{X}}$ is Möbius-$\uparrow$ monotone, see Corollary 3.1 in [LS12].

More examples of SSDs constructed on partially ordered state spaces can be found in [LS16].

5 Relations between monotonicities in Markov chains

Fix a state space $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$ and partial ordering $\preceq$. Recall that by $\mathbf{X} \in \mathcal{P}$ we mean that the chain has the monotonicity property $\mathcal{P}$, which is defined with respect to this given state space and ordering. For example, the implication "if $\mathbf{X} \in \mathcal{P}_1$ then $\mathbf{X} \in \mathcal{P}_2$" means that if $\mathbf{X}$ is $\mathcal{P}_1$-monotone then it is $\mathcal{P}_2$-monotone with respect to *the same* state space and ordering. For a general ordering $\preceq$, we present the relations between the different concepts of monotonicity in Theorem 5.1.

Theorem 5.1. *For a discrete time Markov chain $\mathbf{X}$ on a finite state space $\mathbb{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$ which is partially ordered by $\preceq$, we have the following implications:*

1. *If $\mathbf{X} \in \mathcal{R}$ then $\mathbf{X} \in \mathcal{S}$*
2. *If $\mathbf{X} \in \mathcal{S}$ then $\mathbf{X} \in \mathcal{W}^{\uparrow}$*
3. *If $\mathbf{X} \in \mathcal{S}$ then $\mathbf{X} \in \mathcal{W}^{\downarrow}$*
4. *If $\mathbf{X} \in \mathcal{M}^{\uparrow}$ then $\mathbf{X} \in \mathcal{W}^{\uparrow}$*
5. *If $\mathbf{X} \in \mathcal{M}^{\downarrow}$ then $\mathbf{X} \in \mathcal{W}^{\downarrow}$*

We derive and recall some useful properties of the Möbius function of a partial ordering. First we will show that $\sum_{\mathbf{e}_i \in \mathbb{E}} \mu(\mathbf{e}_i, \mathbf{e}) = 0$ for any poset with the minimum state (we denote it by $\mathbf{e}_1$) and that $\sum_{\mathbf{e}_i \in \mathbb{E}} \mu(\mathbf{e}, \mathbf{e}_i) = 0$ for any poset with the maximum state (denoted by $\mathbf{e}_M$).

It is known (see [Rot64]) that the matrix $\mathbf{C}^{-1} = \mu$ can be calculated recursively:

$$\mu(\mathbf{e}_i, \mathbf{e}_j) = \begin{cases} 1 & \text{if} \quad \mathbf{e}_i = \mathbf{e}_j \\ -\sum_{\mathbf{e}_i \prec \mathbf{e}_k \preceq \mathbf{e}_j} \mu(\mathbf{e}_k, \mathbf{e}_j) & \text{if} \quad \mathbf{e}_i \prec \mathbf{e}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

or by inverting the matrix $\mathbf{C}$ using Gauss–Jordan elimination by columns instead of rows:

$$= \begin{cases} 1 & \text{if} \quad \mathbf{e}_i = \mathbf{e}_j \\ -\sum_{\mathbf{e}_i \preceq \mathbf{e}_k \prec \mathbf{e}_j} \mu(\mathbf{e}_i, \mathbf{e}_k) & \text{if} \quad \mathbf{e}_i \prec \mathbf{e}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Therefore, using (6): for any poset with the minimum state and any state $\mathbf{e}$ which is not the minimum, we have

$$\sum_{\mathbf{e}_i \in \mathbb{E}} \mu(\mathbf{e}_i, \mathbf{e}) = \sum_{\mathbf{e}_i \in \mathbb{E}: \mathbf{e}_i \not\preceq \mathbf{e}} \mu(\mathbf{e}_i, \mathbf{e}) + \sum_{\mathbf{e}_1 \prec \mathbf{e}_i \preceq \mathbf{e}} \mu(\mathbf{e}_i, \mathbf{e}) + \mu(\mathbf{e}_1, \mathbf{e})$$

$$= 0 - \mu(\mathbf{e}_1, \mathbf{e}) + \mu(\mathbf{e}_1, \mathbf{e}) = 0.$$

Similarly, using (7): for any poset with the maximum state and for any state $\mathbf{e}$ which is not the maximum, we have

$$\sum_{\mathbf{e}_i \in \mathbb{E}} \mu(\mathbf{e}, \mathbf{e}_i) = \sum_{\mathbf{e}_i \in \mathbb{E}: \mathbf{e} \not\preceq \mathbf{e}_i} \mu(\mathbf{e}, \mathbf{e}_i) + \sum_{\mathbf{e} \preceq \mathbf{e}_i \prec \mathbf{e}_M} \mu(\mathbf{e}, \mathbf{e}_i) + \mu(\mathbf{e}, \mathbf{e}_M)$$

$$= 0 - \mu(\mathbf{e}, \mathbf{e}_M) + \mu(\mathbf{e}, \mathbf{e}_M) = 0.$$

We will write $\mathbf{e}^+$ for an arbitrary successor of $\mathbf{e}$. For a poset $(\mathbb{E}, \preceq)$ with the maximum $\mathbf{e}_M$ we can consider the subspaces $\{\mathbf{e}\}^\uparrow, \{\mathbf{e}^+\}^\uparrow$ with the minimum states $\mathbf{e}$ and $\mathbf{e}^+$ respectively. From the above consideration, we have

$$\sum_{\mathbf{e}_i : \mathbf{e} \preceq \mathbf{e}_i, \mathbf{e}^+ \not\preceq \mathbf{e}_i} \mu(\mathbf{e}_i, \mathbf{e}') = \sum_{\mathbf{e}_i : \mathbf{e} \preceq \mathbf{e}_i \preceq \mathbf{e}_M} \mu(\mathbf{e}_i, \mathbf{e}') - \sum_{\mathbf{e}_i : \mathbf{e}^+ \preceq \mathbf{e}_i \preceq \mathbf{e}_M} \mu(\mathbf{e}_i, \mathbf{e}') = \tag{8}$$

$$\begin{cases} 1-0 & \text{if } \mathbf{e}' = \mathbf{e}, \\ 0-1 & \text{if } \mathbf{e}' = \mathbf{e}^+ \\ 0-0 & \text{otherwise} \end{cases} = \begin{cases} 1 & \text{if } \mathbf{e}' = \mathbf{e}, \\ -1 & \text{if } \mathbf{e}' = \mathbf{e}^+, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Similarly, for a poset $(\mathbb{E}, \preceq)$ with the minimum state $\mathbf{e}_1$ we can consider the subspaces $\{\mathbf{e}\}^\downarrow, \{\mathbf{e}^+\}^\downarrow$ with the maximum states $\mathbf{e}$ and $\mathbf{e}^+$ respectively. We have:

$$\sum_{\mathbf{e}_i : \mathbf{e}_i \preceq \mathbf{e}^+, \mathbf{e}_i \not\preceq \mathbf{e}} \mu(\mathbf{e}', \mathbf{e}_i) = \sum_{\mathbf{e}_i : \mathbf{e}_1 \preceq \mathbf{e}_i \preceq \mathbf{e}^+} \mu(\mathbf{e}', \mathbf{e}_i) - \sum_{\mathbf{e}_i : \mathbf{e}_1 \preceq \mathbf{e}_i \preceq \mathbf{e}} \mu(\mathbf{e}', \mathbf{e}_i) = \tag{10}$$

$$\begin{cases} 0-1 & \text{if } \mathbf{e}' = \mathbf{e}, \\ 1-0 & \text{if } \mathbf{e}' = \mathbf{e}^+, \\ 0-0 & \text{otherwise} \end{cases} = \begin{cases} -1 & \text{if } \mathbf{e}' = \mathbf{e}, \\ 1 & \text{if } \mathbf{e}' = \mathbf{e}^+, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Proof of Theorem 5.1. Implications 1–3 are known (see, e.g., [FM01]), we include short proofs in the notation of this paper for completeness.

1. We want to show that for any states $\mathbf{e}_i \preceq \mathbf{e}_j$ and any upset U, the inequality $\mathbf{P}(\mathbf{e}_i, U) \leqslant \mathbf{P}(\mathbf{e}_j, U)$ is fulfilled. From the definition of update function and the property of realizable monotonicity (3), we have

   $$\mathbf{P}(\mathbf{e}_i, U) = \int_0^1 \mathbb{1}(\phi(\mathbf{e}_i, u) \in U) du \leqslant \int_0^1 \mathbb{1}(\phi(\mathbf{e}_j, u) \in U) du = \mathbf{P}(\mathbf{e}_j, U)$$

   as the indicator of set is equal to 1 when the indicator of its subset is equal to 1.

2. & 3. For any state $\mathbf{e}$, $\{\mathbf{e}\}^\uparrow$ is an upset and $\{\mathbf{e}\}^\downarrow$ is a downset. Thus stochastic monotonicity implies that the monotonicity is preserved for any $\{\mathbf{e}\}^\uparrow$ and $\{\mathbf{e}\}^\downarrow$, which is the definition of $^\uparrow$-weak and $^\downarrow$-weak monotonicity.

4. & 5. Möbius-$^\downarrow$ monotonicity means

   $$\forall_{\mathbf{e} \preceq \mathbf{e}_k} \sum_{\mathbf{e}_i : \mathbf{e} \preceq \mathbf{e}_i} \mu(\mathbf{e}, \mathbf{e}_i) \mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^\downarrow) \geqslant 0,$$

   thus for arbitrary $\mathbf{e}_j \in \mathbb{E}$ we have

   $$\sum_{\mathbf{e} : \mathbf{e}_j \preceq \mathbf{e}, \mathbf{e}_j^+ \not\preceq \mathbf{e}} \sum_{\mathbf{e}_i : \mathbf{e} \preceq \mathbf{e}_i} \mu(\mathbf{e}, \mathbf{e}_i) \mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^\downarrow) \geqslant 0.$$

   Changing the order of summation we have

   $$\sum_{\mathbf{e}_i : \mathbf{e}_j \preceq \mathbf{e}_i} \sum_{\mathbf{e} : \mathbf{e} \preceq \mathbf{e}_i, \mathbf{e}_j \preceq \mathbf{e}, \mathbf{e}_j^+ \not\preceq \mathbf{e}} \mu(\mathbf{e}, \mathbf{e}_i) \mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^\downarrow) \geqslant 0,$$

   $$\sum_{\mathbf{e}_i : \mathbf{e}_i \succeq \mathbf{e}_j} \mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^\downarrow) \sum_{\mathbf{e} : \mathbf{e} \preceq \mathbf{e}_i, \mathbf{e}_j \preceq \mathbf{e}, \mathbf{e}_j^+ \not\preceq \mathbf{e}} \mu(\mathbf{e}, \mathbf{e}_i) \geqslant 0.$$

   Using (9) for each $\mathbf{e}_i$ for the (sub-)poset $(\{\mathbf{e}_i\}^\downarrow, \preceq)$ (where $\mathbf{e}_i$ is the maximum) with its subspaces $\{\mathbf{e}\}^\uparrow, \{\mathbf{e}^+\}^\uparrow$, we have

   $$\mathbf{P}(\mathbf{e}_j, \{\mathbf{e}_k\}^\downarrow) - \mathbf{P}(\mathbf{e}_j^+, \{\mathbf{e}_k\}^\downarrow) \geqslant 0$$

   for any $\mathbf{e}_j, \mathbf{e}_k$.
   The proof that Möbius-$^\uparrow$ monotonicity implies weak-$^\uparrow$ is similar.

□

In Fig. 1, Theorem 5.1 is summarized.

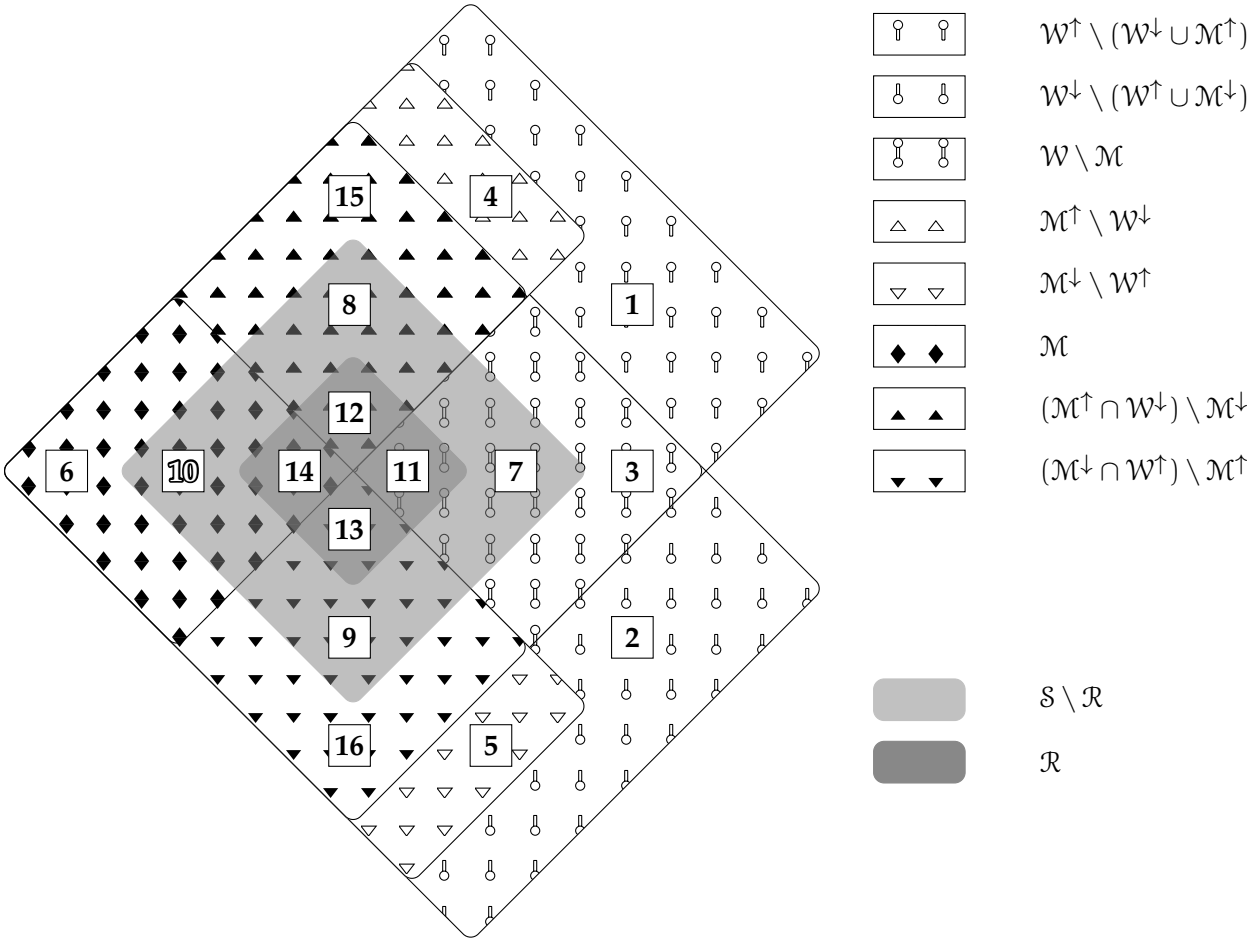


Figure 1: Relation between monotonicities. General partial ordering.

We know of no other implications involving the monotonicities we have considered. More precisely, the only one we do not know is whether Möbius-↑, Möbius-↓ and stochastic monotonicities imply realizable monotonicity (which is stated below as an open problem). The nonexistence of other implications is proven by presenting examples in Appendix A (the numbers in Fig. 1 correspond to the enumeration of these examples).

Open problem 1. *Does there exist a chain* $\mathbf{X}$*, state space* $\mathbb{E}$*, and partial ordering* $\preceq$ *such that* $\mathbf{X} \in \mathcal{M}^{\downarrow} \cap \mathcal{M}^{\uparrow} \cap \mathcal{S} \setminus \mathcal{R}$ *?*

Remark 5.2 (On Möbius monotonicity)**.** For a total ordering (denote the states by $\mathbb{E} = \{1, \ldots, M\}$), the stochastic monotonicity of $\mathbf{X}$ can be written as

$$\forall (j, i_1 \leqslant i_2) \quad \mathbf{P}_X(i_1, \{j\}^{\uparrow}) \leqslant \mathbf{P}_X(i_2, \{j\}^{\uparrow}) \quad \equiv \quad \mathbf{P}_X(i_1, \{j\}^{\downarrow}) \geqslant \mathbf{P}_X(i_2, \{j\}^{\downarrow}).$$

In this ordering we can think of this monotonicity in two different (though equivalent) ways:

- *"Understanding 1"*. For any upset U and $\forall (i_1 \leqslant i_2)$ we have $\mathbf{P}_X(i_1, U) \leqslant \mathbf{P}_X(i_2, U)$ (or equivalently: for any downset D and $\forall (i_1 \leqslant i_2)$ we have $\mathbf{P}_X(i_1, D) \geqslant \mathbf{P}_X(i_2, D)$).

- *"Understanding 2"*. For any upset U define $F_U(i) := \mathbf{P}_X(i, U)$. Then $\mathbf{X}$ is stochastically monotone if the function $F_U(i)$, treated as a function of i, must be "like" a distribution function, i.e., $\forall(i_1 \leqslant i_2) F_U(i_1) \leqslant F_U(i_2)$.

  Equivalently: for any downset D define $\bar{F}_D(i) = \mathbf{P}_X(i, D)$. Then $\mathbf{X}$ is stochastically monotone if the function $\bar{F}_D(i)$, treated as a function of i, must be "like" the tail of a distribution function, i.e., $\forall(i_1 \leqslant i_2) \bar{F}_D(i_1) \geqslant \bar{F}_D(i_2)$.

Extending *"Understanding 1"* to a partial ordering $\preceq$ (we simply have different downsets and upsets, and each $i_1 \leqslant i_2$ is replaced by $\mathbf{e} \preceq \mathbf{e}'$) leads to stochastic monotonicity as defined in Definition 3.1. Extending *"Understanding 2"* with $F_U(\cdot)$ being like a distribution function ($\bar{F}_D(\cdot)$ being like the tail of a distribution function) leads to Möbius-$^\downarrow$ (Möbius-$^\uparrow$) monotonicity, as defined in Definition 3.7.

5.1 Tree-ordering and total ordering

For a general partial ordering, we have, in Theorem 5.1, determined all the monotonicity relations. In this section we restrict our attention to some special cases: tree ordering and linear ordering.

Tree ordering. Let us start with a definition of this ordering.

Definition 5.3. *A partial ordering $\preceq$ on $\mathbb{E}$ is called a* tree ordering *if there exists a maximum (which has no predecessor) and every other (non-maximal) state $\mathbf{e}$ has exactly one predecessor.*

This definition affords a straightforward algorithm for inverting the matrix $\mathbf{C} = \mathbb{1}(\mathbf{e}_i \preceq \mathbf{e}_j)$. For a column corresponding to the state $\mathbf{e}$, it is enough to subtract the columns corresponding to the successors of $\mathbf{e}$. We obtain the matrix

$$\mu(\mathbf{e}_i, \mathbf{e}_j) = \begin{cases} 1 & \text{if} \quad \mathbf{e}_i = \mathbf{e}_j, \\ -1 & \text{if} \quad \mathbf{e}_i^+ = \mathbf{e}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Theorem 5.4. *Let $\mathbf{X}$ be a Markov chain on $\mathbb{E}$ with a tree ordering $\preceq$. Then the following statements are equivalent.*

(i) $\mathbf{X} \in \mathcal{S}$

(ii) $\mathbf{X} \in \mathcal{R}$

(iii) $\mathbf{X} \in \mathcal{M}^\downarrow$

(iv) $\mathbf{X} \in \mathcal{W}^\downarrow$

Proof.

- $(i) \iff (ii)$

  Implication $(ii) \Rightarrow (i)$ follows from implication 1 of Theorem 5.1, whereas $(i) \Rightarrow (ii)$ follows from Theorem 4.3 in [FM01].

- $(iii) \iff (iv)$

  Implication $(iii) \Rightarrow (iv)$ follows from implication 5 of Theorem 5.1. To show $(iv) \Rightarrow (iii)$ we assume Möbius-$^\downarrow$ monotonicity, i.e., we have (the Möbius function given in Eq. (12))

  $$\forall(\mathbf{e}_i, \mathbf{e}_j \in \mathbb{E}) \qquad 0 \leqslant \mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_j\}^\downarrow) - \mathbf{P}(\mathbf{e}_i^+, \{\mathbf{e}_j\}^\downarrow) = \sum_{\mathbf{e} \succeq \mathbf{e}_i} \mu(\mathbf{e}_i, \mathbf{e}) \mathbf{P}(\mathbf{e}, \{\mathbf{e}_j\}^\downarrow),$$

  which is exactly weak-$^\downarrow$ monotonicity.

- $(i) \iff (iv)$

  Implication $(i) \Rightarrow (iv)$ follows from implication 3 of Theorem 5.1. To show $(iv) \Rightarrow (i)$, note that any downset D can be written as a disjoint union of sets of the form $\{\mathbf{e}_k\}^{\downarrow}$, i.e., $D = \bigcup_{k \in K} \{\mathbf{e}_k\}^{\downarrow}$ for some $K \subseteq \mathbb{E}$. For any $\mathbf{e}_k$, weak-$^{\downarrow}$ monotonicity implies that $\mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^{\downarrow}) - \mathbf{P}(\mathbf{e}_i^+, \{\mathbf{e}_k\}^{\downarrow}) \geqslant 0$, thus

$$\mathbf{P}(\mathbf{e}_i, D) - \mathbf{P}(\mathbf{e}_i^+, D) = \sum_{k \in K} \left(\mathbf{P}(\mathbf{e}_i, \{\mathbf{e}_k\}^{\downarrow}) - \mathbf{P}(\mathbf{e}_i^+, \{\mathbf{e}_k\}^{\downarrow}) \right) \geqslant 0,$$

  which implies stochastic monotonicity.

□

The monotonicity relations for tree-ordering are summarized in Fig. 2. The examples numbered 17, 18, 19 and 20 are given in Appendix A.

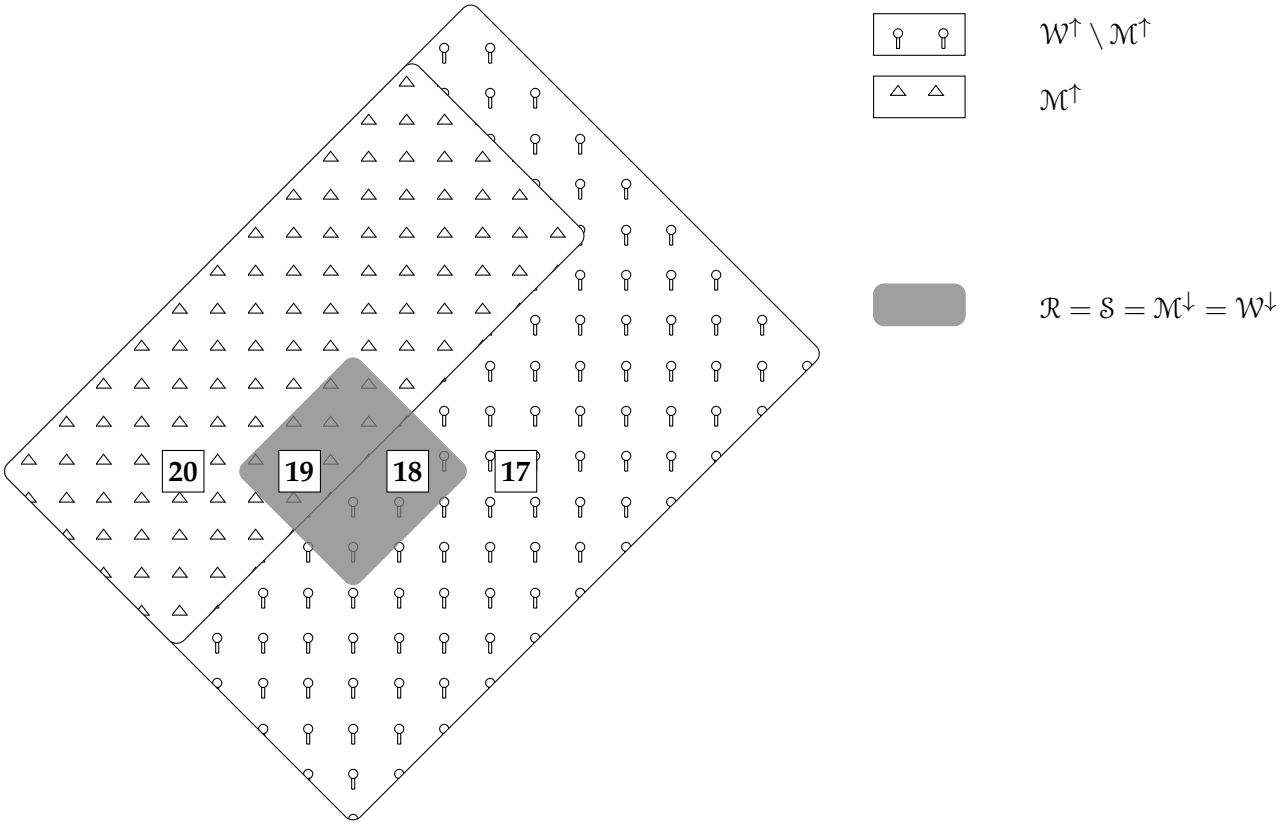


Figure 2: Relation between monotonicities. Tree-like ordering.

Total ordering. For this ordering let us denote the elements of state space $\mathbb{E}$ by $\{1, \ldots, M\}$. The Möbius function is following:

$$\mu(i,j) = \begin{cases} 1 & \text{if} \quad j = i, \\ -1 & \text{if} \quad j = i+1, \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

with ones on the diagonal and minus ones directly above it. For this ordering, we have the following lemma.

Lemma 5.5. *Let* $\mathbf{X}$ *be a Markov chain on* $\mathbb{E}$ *with total ordering* $\preceq := \leqslant$. *Then all the monotonicites* $\mathcal{S}, \mathcal{R}, \mathcal{M}^{\uparrow}, \mathcal{M}^{\downarrow}, \mathcal{W}^{\uparrow}, \mathcal{W}^{\downarrow}$ *are equivalent.*

Proof. By Theorem 5.4 it is enough to show that $\mathcal{W}^{\uparrow}$ is equivalent to $\mathcal{W}^{\downarrow}$ and that $\mathcal{W}^{\uparrow} \Rightarrow \mathcal{M}^{\uparrow}$. For a total ordering, all upsets are of the form $\{k\}^{\uparrow} = \{k, \ldots, M\}$ and all downsets are of the form $\{k\}^{\downarrow} = \{1, \ldots, k\}$. Hence (since the complement of a downset is an upset and vice versa), they are equivalent to each other and actually denote stochastic monotonicity.
Note that

$$\sum_{k \in \mathbb{E}} \mu(k, i)\mathbf{P}(k, \{j\}^{\uparrow}) = \mathbf{P}(i+1, \{j\}^{\uparrow}) - \mathbf{P}(i, \{j\}^{\uparrow}),$$

which means that $\mathcal{W}^{\uparrow}$ and $\mathcal{M}^{\uparrow}$ are equivalent.

□

6 Monotonicities and Siegmund duality

As mentioned, strong stationary duality was introduced in [DF90b]. However, a somewhat general recipe for such an SSD was only given in the case when the time reversal was stochastically monotone with respect to a total ordering. In Section 4.3 we recalled the theorem from [LS12], that for a given ergodic chain $\mathbf{X}$ there exists a strong stationary dual chain $\mathbf{X}^*$ (with the link being a truncated stationary distribution) if and only if the time reversal of $\mathbf{X}$ is Möbius monotone. It turns out that there is a close connection between SSD and another duality. We say that the chain $\mathbf{Z}$ is a **Siegmund dual** of $\mathbf{X}$ if for any $n \geqslant 0, \mathbf{e}_i, \mathbf{e}_j \in \mathbb{E}$ we have $\Pr(X_n \preceq \mathbf{e}_j | X_0 = \mathbf{e}_i) = \Pr(Z_n \succeq \mathbf{e}_i | Z_0 = \mathbf{e}_j)$. Siegmund [Sie76] showed that for a total ordering, such a dual exists if and only if $\mathbf{X}$ is stochastically monotone. Lorek [Lor] gives an extension to partial orderings (the existence of the minimum and the maximum is required). The main result is that the Siegmund dual exists if and only if the chain is **Möbius**-$^{\downarrow}$ monotone. Moreover, in the latter article it is shown that the SSD from [LS12] can be constructed in the following three steps: i) Calculate the time reversal of $\mathbf{X}$; ii) Calculate its Siegmund dual; iii) Calculate the appropriate Doob h-transform.

The results of this article are relevant for SSD and Siegmund duality. The general constructions of the SSD and the Siegmund dual were unknown for partial orderings. For Siegmund duality, for partially ordered state spaces, it was known that stochastic monotonicity is "not enough." Liggett, in [Lig04] a book on particle systems) writes (p. 87) "having a (reasonable) dual is a much more special property than being monotone, when the state space is not totally ordered." However, we can obtian an SSD or a Siegmund dual for a chain which is not stochastically monotone, such as is shown with the chain with the transition matrix $\mathbf{P}_6$ in Appendix A (the chain is not stochastically monotone, but is both Möbius-$^{\downarrow}$ and Möbius-$^{\uparrow}$ monotone).

References

[AD86] David Aldous and Persi Diaconis. Shuffling cards and stopping times. *American Mathematical Monthly*, 93(5):333–348, 1986.

[CLR01] George Casella, Michael Lavine, and Christian P. Robert. Explaining the Perfect Sampler. *The American Statistician*, 55(4):299–305, 2001.

[Com70] Louis Comtet. *Analyse combinatoire*. Presses universitaires de France, Paris, 1970.

[DF90a] Persi Diaconis and James Allen Fill. Examples for the Theory of Strong Stationary Duality with Countable State Spaces. *Probability in the Engineering and Informational Sciences*, 4(2):157–180, 1990.

[DF90b] Persi Diaconis and James Allen Fill. Strong stationary times via a new form of duality. *The Annals of Probability*, 18(4):1483–1522, 1990.

[Dim01] Xeni K. Dimakos. A Guide to Exact Simulation. *International Statistical Review*, 69(1):27–48, 2001.

[DSC06] Persi Diaconis and L Saloff-Coste. Separation cut-offs for birth and death chains. *The Annals of Applied Probability*, 16(4):2098–2122, 2006.

[Fil98] James Allen Fill. An interruptible algorithm for perfect sampling via Markov chains. *The Annals of Applied Probability*, 8(1):131–162, 1998.

[Fil09a] James Allen Fill. On hitting times and fastest strong stationary times for skip-free and more general chains. *Journal of Theoretical Probability*, 22(3):587–600, 2009.

[Fil09b] James Allen Fill. The Passage Time Distribution for a Birth-and-Death Chain: Strong Stationary Duality Gives a First Stochastic Proof. *Journal of Theoretical Probability*, 22(3):543–557, 2009.

[FL14] James Allen Fill and Vince Lyzinski. Hitting times and interlacing eigenvalues: a stochastic approach using intertwinings. *Journal of Theoretical Probability*, 27(3):954–981, 2014.

[FM01] James Allen Fill and Motoya Machida. Stochastic monotonicity and realizable monotonicity. *The Annals of Probability*, 29(2):938–978, 2001.

[FMMR00] James Allen Fill, Motoya Machida, Duncan J. Murdoch, and Jeffrey S. Rosenthal. Extension of Fill's perfect rejection sampling algorithm to general chains. *Random Structures and Algorithms*, 17(3-4):290–316, 2000.

[HM16] Thierry Huillet and Servet Martínez. On Möbius Duality and Coarse-Graining. *Journal of Theoretical Probability*, 29(1):143–179, 2016.

[HN98] O Häggström and K Nelander. Exact sampling from antimonotone systems. *Statistica Neerlandica*, 52(3):360–380, 1998.

[Hub03] Mark Huber. A Bounding Chain for Swendsen-Wang. *Random Structures and Algorithms*, 22(1):43–59, 2003.

[Ken98] Wilfrid S. Kendall. Perfect Simulation for the Area-Interaction Point Process. In L. Accardi and C. C. Heyde, editors, *Probability towards 2000*, volume 128 of *Lecture Notes in Statistics*, pages 218–234. Springer-Verlag, Berlin, 1998.

[KKO77] T. Kamae, U. Krengel, and G. L. O'Brien. Stochastic Inequalities on Partially Ordered Spaces. *The Annals of Probability*, 5(6):899–912, 1977.

[KM00] Wilfrid S. Kendall and Jesper Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32(3):844–865, 2000.

[Lig04] Thomas M. Liggett. *Interacting Particle Systems*. Springer-Verlag, Berlin, Heidelberg, 2004.

[LM17] Paweł Lorek and Piotr Markowski. `The Julia Language` module for checking monotonicities in Markov chains. *GitHub repository,* `https://github.com/lorek/MarkovChains_monotonicities/`. 2017.

[Lor] Paweł Lorek. Siegmund duality for Markov chains on partially ordered state spaces. *To appear in Probability in the Engineering and Informational Sciences*.

[Lor17] Paweł Lorek. Generalized Gambler's Ruin Problem: Explicit Formulas via Siegmund Duality. *Methodology and Computing in Applied Probability*, 19(2):603–613, 2017.

[LS12] Paweł Lorek and Ryszard Szekli. Strong stationary duality for Möbius monotone Markov chains. *Queueing Systems*, 71(1-2):79–95, 2012.

[LS16] Paweł Lorek and Ryszard Szekli. Strong stationary duality for Möbius monotone Markov chains: examples. *Probability and Mathematical Statistics*, 36(1):75–97, 2016.

[Mac01] Motoya Machida. *Stochastic monotonicity and realizable monotonicity*. Phd Thesis, The Johns Hopkins University, 2001.

[PW96] James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[Rot64] Gian-Carlo Rota. On the foundations of combinatorial theory I. Theory of Möbius functions. *Probability Theory and Related Fields*, 368:340–368, 1964.

[Sie76] David Siegmund. The Equivalence of Absorbing and Reflecting Barrier Problems for Stochastically Monotone Markov Processes. *The Annals of Probability*, 4(6):914–924, 1976.

[Str65] V. Strassen. The Existence of Probability Measures with Given Marginals. *The Annals of Mathematical Statistics*, 36(2):423–439, 1965.

A Examples

The relations between monotonicities were given in Theorem 5.1 for a general partial ordering and in Theorem 5.4 for a tree ordering. They were summarized in Figures 1 and 2 respectively. In this section we prove (except for Open problem 1) that all the intersections in these figures are non-empty.
Given $\mathbf{P}$ and $\mathbf{C}$, checking all monotonicities except realizable monotonicity is straightforward (it only requires some matrix operations):

- Checking Möbius-$^{\downarrow}$ and Möbius-$^{\uparrow}$ monotonicity is straightforward from Definition 3.7.
- For weak monotonicities we need to precompute the offspring matrix $\mathbf{R}$. Let $\mathbb{O}(\mathbf{e}_i) = \{\mathbf{e} : \mathbf{e}_i \prec \mathbf{e}$ and $!\exists_{\mathbf{e}_j}(\mathbf{e}_i \prec \mathbf{e}_j \prec \mathbf{e})\}$ be the set of offspring of the state $\mathbf{e}_i$. The offspring matrix is defined as $\mathbf{R} = (\mathbf{R}_{\mathbf{e}_1}^{\mathsf{T}}, \ldots, \mathbf{R}_{\mathbf{e}_M}^{\mathsf{T}})^{\mathsf{T}}$, where $\mathbf{R}_{\mathbf{e}_i}$ is the $|\mathbb{O}(\mathbf{e}_i)| \times |\mathbb{E}|$ matrix such that $\mathbf{R}_{\mathbf{e}_i}(\mathbf{e}_j, \mathbf{e}_i) = 1$ and $\mathbf{R}_{\mathbf{e}_i}(\mathbf{e}_j, \mathbf{e}_j) = -1$ for $\mathbf{e}_j \in \mathbb{O}(\mathbf{e}_i)$, all other entries being equal to zero. Note that $\mathbb{O}(\mathbf{e}_i)$ for all $\mathbf{e}_i \in \mathbb{E}$ and thus the matrix $\mathbf{R}$ is computed from $\mathbf{C}$. Weak-$^{\uparrow}$ monotonicity means that all entries of $\mathbf{RPC}^{\mathsf{T}}$ are nonpositive, whereas weak-$^{\downarrow}$ means that all entries of $\mathbf{RPC}$ nonnegative.
- For stochastic monotonicity, we additionally need the matrix of all upsets (denoted by $\mathbf{S}$) instead of "just" the ordering matrix $\mathbf{C}$ ($\mathbf{S}$ is computed from $\mathbf{C}$). Stochastic monotonicity means that all entries of $\mathbf{RPS}$ are nonnegative.

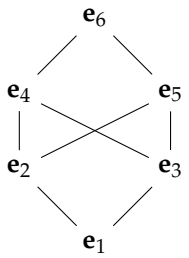
We wrote functions in `The Julia Language` [LM17] checking all the above monotonicities. The functions require the transition matrix $\mathbf{P}$ and the ordering matrix $\mathbf{C}$. Also, the script checking the above monotonicities of all the examples that follow is available. The proofs concerning realizable monotonicites are given after introducing the examples. Recall that for a tree ordering, realizable monotonicity is equivalent to (among others) stochastic monotonicity, thus checking the latter is enough.

The pairing of the ordering matrices and the transition matrices:

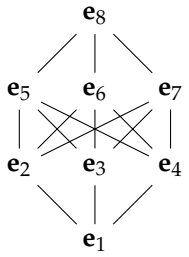
- $\mathbf{C}_1$ for processes $\mathbf{P}_3, \mathbf{P}_6, \mathbf{P}_7, \mathbf{P}_{11}, \mathbf{P}_{12}, \mathbf{P}_{13}, \mathbf{P}_{14}, \mathbf{P}_{15}, \mathbf{P}_{16}$,
- $\mathbf{C}_2$ for $\mathbf{P}_1, \mathbf{P}_2$,
- $\mathbf{C}_3$ for $\mathbf{P}_4, \mathbf{P}_5, \mathbf{P}_8, \mathbf{P}_9$ and
- $\mathbf{C}_4$ for $\mathbf{P}_{17}, \mathbf{P}_{18}, \mathbf{P}_{19}, \mathbf{P}_{20}$.

Order matrices and Hasse diagrams:

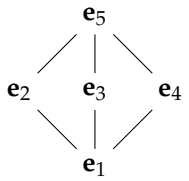
$$\mathbf{C}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



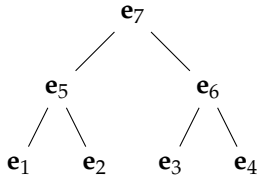
$$\mathbf{C}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathbf{C}_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathbf{C}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



The transition matrices

1. $\mathcal{W}^{\uparrow} \setminus (\mathcal{W}^{\downarrow} \cup \mathcal{M}^{\uparrow})$

$$\mathbf{P}_1 = \begin{pmatrix} 3/8 & 1/8 & 1/8 & 1/8 & 0 & 1/8 & 1/8 & 0 \\ 1/4 & 1/4 & 0 & 1/8 & 1/8 & 0 & 1/8 & 1/8 \\ 1/4 & 0 & 3/8 & 0 & 0 & 1/4 & 0 & 1/8 \\ 1/4 & 0 & 1/8 & 0 & 1/8 & 1/4 & 1/8 & 1/8 \\ 0 & 1/8 & 1/8 & 0 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 1/8 & 0 & 3/8 \\ 0 & 0 & 1/8 & 1/8 & 1/4 & 1/4 & 1/8 & 1/8 \\ 0 & 0 & 1/8 & 0 & 1/8 & 1/8 & 0 & 5/8 \end{pmatrix}$$

2. $\mathcal{W}^{\downarrow} \setminus (\mathcal{W}^{\uparrow} \cup \mathcal{M}^{\downarrow})$

$$\mathbf{P}_2 = \begin{pmatrix} 5/8 & 0 & 1/8 & 1/8 & 0 & 1/8 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 1/4 & 1/8 & 1/8 & 0 & 0 \\ 3/8 & 0 & 1/8 & 0 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 0 & 0 & 1/8 & 1/8 & 0 \\ 1/8 & 1/8 & 1/4 & 1/8 & 0 & 1/8 & 0 & 1/4 \\ 1/8 & 0 & 1/4 & 0 & 0 & 3/8 & 0 & 1/4 \\ 1/8 & 1/8 & 0 & 1/8 & 1/8 & 0 & 1/4 & 1/4 \\ 0 & 1/8 & 1/8 & 0 & 1/8 & 1/8 & 1/8 & 3/8 \end{pmatrix}$$

3. $\mathcal{W} \setminus (\mathcal{M} \cup \mathcal{S})$

$$\mathbf{P}_3 = \begin{pmatrix} 1/2 & 1/6 & 0 & 1/3 & 0 & 0 \\ 1/3 & 1/6 & 1/6 & 1/3 & 0 & 0 \\ 1/3 & 1/6 & 0 & 1/3 & 1/6 & 0 \\ 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/3 \\ 1/6 & 0 & 1/6 & 1/6 & 1/3 & 1/6 \\ 0 & 1/6 & 1/3 & 0 & 1/6 & 1/3 \end{pmatrix}$$

4. $\mathcal{M}^{\uparrow} \setminus \mathcal{W}^{\downarrow}$

$$\mathbf{P}_4 = \begin{pmatrix} 2/5 & 1/5 & 1/5 & 1/5 & 0 \\ 2/5 & 1/5 & 1/5 & 1/5 & 0 \\ 0 & 2/5 & 2/5 & 1/5 & 0 \\ 0 & 2/5 & 1/5 & 2/5 & 0 \\ 0 & 1/5 & 2/5 & 0 & 2/5 \end{pmatrix}$$

5. $\mathcal{M}^{\downarrow} \setminus \mathcal{W}^{\uparrow}$

$$\mathbf{P}_5 = \begin{pmatrix} 2/5 & 0 & 2/5 & 1/5 & 0 \\ 0 & 2/5 & 1/5 & 2/5 & 0 \\ 0 & 1/5 & 2/5 & 2/5 & 0 \\ 0 & 1/5 & 1/5 & 1/5 & 2/5 \\ 0 & 1/5 & 1/5 & 1/5 & 2/5 \end{pmatrix}$$

6. $\mathcal{M} \setminus \mathcal{S}$

$$\mathbf{P}_6 = \begin{pmatrix} 17/24 & 0 & 0 & 1/8 & 1/8 & 1/24 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/24 & 1/8 & 1/8 & 0 & 0 & 17/24 \end{pmatrix}$$

7. $\mathcal{S} \setminus (\mathcal{M}^{\uparrow} \cup \mathcal{M}^{\downarrow} \cup \mathcal{R})$

$$\mathbf{P}_7 = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/3 \\ 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/3 \end{pmatrix}$$

8. $\mathcal{S} \cap \mathcal{M}^{\uparrow} \setminus (\mathcal{M}^{\downarrow} \cup \mathcal{R})$

$$\mathbf{P}_8 = \begin{pmatrix} 2/5 & 1/5 & 1/5 & 1/5 & 0 \\ 2/5 & 1/5 & 1/5 & 1/5 & 0 \\ 2/5 & 0 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 2/5 & 1/5 & 0 \\ 0 & 2/5 & 1/5 & 0 & 2/5 \end{pmatrix}$$

9. $\mathcal{S} \cap \mathcal{M}^{\downarrow} \setminus (\mathcal{M}^{\uparrow} \cup \mathcal{R})$

$$\mathbf{P}_9 = \begin{pmatrix} 2/5 & 0 & 1/5 & 2/5 & 0 \\ 0 & 1/5 & 2/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 0 & 2/5 \\ 0 & 1/5 & 1/5 & 1/5 & 2/5 \\ 0 & 1/5 & 1/5 & 1/5 & 2/5 \end{pmatrix}$$

10. Aforementioned open problem.

11. $\mathcal{R} \setminus (\mathcal{M}^{\uparrow} \cup \mathcal{M}^{\downarrow})$

$$\mathbf{P}_{11} = \begin{pmatrix} 1/3 & 1/6 & 1/6 & 1/6 & 1/6 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/3 \end{pmatrix}$$

12. $\mathcal{R} \cap \mathcal{M}^{\uparrow} \setminus \mathcal{M}^{\downarrow}$

$$\mathbf{P}_{12} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 1/3 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

13. $\mathcal{R} \cap \mathcal{M}^{\downarrow} \setminus \mathcal{M}^{\uparrow}$

$$\mathbf{P}_{13} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/3 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

14. $\mathcal{R} \cap \mathcal{M}$

$$\mathbf{P}_{14} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

15. $\mathcal{W}^{\downarrow} \cap \mathcal{M}^{\uparrow} \setminus (\mathcal{M}^{\downarrow} \cup \mathcal{S})$

$$\mathbf{P}_{15} = \begin{pmatrix} 17/24 & 0 & 0 & 1/8 & 1/8 & 1/24 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/24 & 1/16 & 1/16 & 1/16 & 1/16 & 17/24 \end{pmatrix}$$

16. $\mathcal{W}^{\uparrow} \cap \mathcal{M}^{\downarrow} \setminus (\mathcal{M}^{\uparrow} \cup \mathcal{S})$

$$\mathbf{P}_{16} = \begin{pmatrix} 17/24 & 1/16 & 1/16 & 1/16 & 1/16 & 1/24 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/8 & 5/16 & 5/16 & 1/12 & 1/12 & 1/12 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/12 & 1/12 & 1/12 & 5/16 & 5/16 & 1/8 \\ 1/24 & 1/8 & 1/8 & 0 & 0 & 17/24 \end{pmatrix}$$

Examples 17–20 deal with tree-ordering.

17. $\mathcal{W}^{\uparrow} \setminus \mathcal{S}$

$$\mathbf{P_{17}} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}.$$

18. $\mathcal{S} \setminus \mathcal{M}^{\uparrow}$

$$\mathbf{P_{18}} = \begin{pmatrix} 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \end{pmatrix}.$$

19. $\mathcal{W}^{\uparrow} \cap \mathcal{S} \cap \mathcal{M}^{\uparrow}$

$$\mathbf{P_{19}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

20. $\mathcal{M}^{\uparrow} \setminus \mathcal{S}$

$$\mathbf{P_{20}} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

As already stated, checking all monotonicities except realizable monotonicity can be done automatically on a computer. Note that if the chain is not stochastically monotone, then it cannot be realizable monotone. That is why we only need to prove that

- The chains with the transition matrices $\mathbf{P}_{11}, \mathbf{P}_{12}, \mathbf{P}_{13}, \mathbf{P}_{14}$ are realizable monotone.
- The chains with the transition matrices $\mathbf{P}_7, \mathbf{P}_8, \mathbf{P}_9$ are not realizable monotone.

To prove realizable monotonicity it is enough to provide a monotone update rule, whereas showing that a given chain is not realizable monotone (i.e., that no monotone update function exists) is more challenging.

Monotone update rules for the chains with the transition matrices $\mathbf{P}_{11}, \mathbf{P}_{12}, \mathbf{P}_{13}$ and $\mathbf{P}_{14}$. For $\bigcup_{i=1}^{6} A_i = [0,1]$ and $P(U \in A_i) = 1/6$ for $i = 1, \ldots, 6$ and $U \sim Unif[0,1]$ the following functions are monotone w.r.t. the partial ordering defined by $\mathbf{C}_1$.

- $\mathbf{P}_{11}$

$$\begin{aligned} \phi(\mathbf{e}_j, u) &= \mathbf{e}_i \quad \text{if } u \in A_i, \text{for } i = 1, \ldots, 6, j = 2, \ldots, 5, \\ \phi(\mathbf{e}_1, u) &= \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_6, \\ \mathbf{e}_i & \text{if} \quad u \in A_i, i = 1, \ldots, 5, \end{cases} \\ \phi(\mathbf{e}_6, u) &= \begin{cases} \mathbf{e}_6 & \text{if} \quad u \in A_1, \\ \mathbf{e}_i & \text{if} \quad u \in A_i, i = 2, \ldots, 6. \end{cases} \end{aligned}$$

- $\mathbf{P}_{12}$

$$\begin{aligned} \phi(\mathbf{e}_j, u) &= \mathbf{e}_i \quad \text{if } u \in A_i, \text{for } i = 1, \ldots, 6, j = 1, \ldots, 5, \\ \phi(\mathbf{e}_6, u) &= \begin{cases} \mathbf{e}_6 & \text{if} \quad u \in A_1, \\ \mathbf{e}_i & \text{if} \quad u \in A_i, i = 2, \ldots, 6. \end{cases} \end{aligned}$$

- $\mathbf{P}_{13}$

$$\begin{aligned}\phi(\mathbf{e}_j, u) &= \mathbf{e}_i \quad \text{if } u \in A_i, \text{for } i = 1,\dots,6, j = 2,\dots,6,\\ \phi(\mathbf{e}_1, u) &= \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_6,\\ \mathbf{e}_i & \text{if} \quad u \in A_i, i = 1,\dots,5.\end{cases}\end{aligned}$$

- $\mathbf{P}_{14}$

$$\phi(\mathbf{e}_j, u) = \mathbf{e}_i \quad \text{if } u \in A_i, \text{for } i = 1,\dots,6, j = 1,\dots,6.$$

Proofs that $\mathbf{P}_7, \mathbf{P}_8, \mathbf{P}_9$ are not realizable monotone w.r.t. the partial ordering defined by $\mathbf{C}_1$.

- The transition matrix $\mathbf{P}_7$.

  The idea of the proof is the following: we try to construct a monotone update function ϕ and deduce a contradiction. Start with defining an arbitrary update function at state $\mathbf{e}_1$:

$$\phi(\mathbf{e}_1, u) = \mathbf{e}_i \quad \text{if } u \in A_i, i = 1, 2, 3,$$

  for $\bigcup_{i=1}^{3} A_i = [0,1]$, $P(U \in A_i) = 1/3$, $i = 1,2,3$ and $U \sim \mathtt{Unif}[0,1]$. Since $\mathbf{e}_1 \preceq \mathbf{e}_2$ we have the following requirements for $\phi(\mathbf{e}_2, \cdot)$: namely $\phi(\mathbf{e}_2, u) \succeq \mathbf{e}_i$ for $u \in A_i, i = 1,2,3$. Thus the function is uniquely determined:

$$\phi(\mathbf{e}_2, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_1,\\ \mathbf{e}_2 & \text{if} \quad u \in A_2,\\ \mathbf{e}_4 & \text{if} \quad u \in A_3.\end{cases}$$

  Similarly, since $\mathbf{e}_1 \preceq \mathbf{e}_3$, we conclude that

$$\phi(\mathbf{e}_3, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_1,\\ \mathbf{e}_3 & \text{if} \quad u \in A_3,\\ \mathbf{e}_4 & \text{if} \quad u \in A_2.\end{cases}$$

  Also, since $\mathbf{e}_2 \preceq \mathbf{e}_4$, we conclude that

$$\phi(\mathbf{e}_4, u) = \begin{cases} \mathbf{e}_2 & \text{if} \quad u \in A_2,\\ \mathbf{e}_3 & \text{if} \quad u \in A_1,\\ \mathbf{e}_4 & \text{if} \quad u \in A_3.\end{cases}$$

  But this function is not monotone, since for $u \in A_2$ we have $\phi(\mathbf{e}_3, u) = \mathbf{e}_4 \not\preceq \mathbf{e}_2 = \phi(\mathbf{e}_4, u)$.

- The transition matrices $\mathbf{P}_8$ and $\mathbf{P}_9$.

  The idea of the proof is similar to the previous case. It will be done only for $\mathbf{P}_8$ (the proof for $\mathbf{P}_9$ is almost identical, since $\mathbf{P}_9(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{P}_8(\mathbf{e}_{6-i}, \mathbf{e}_{6-j}), i, j = 1,\dots,5$).

  We can start with defining an arbitrary update function at state $\mathbf{e}_1$:

$$\phi(\mathbf{e}_1, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_0,\\ \mathbf{e}_i & \text{if} \quad u \in A_i, i = 1,2,3,4\end{cases}$$

  for $\bigcup_{i=0}^{4} A_i = [0,1]$, $P(U \in A_i) = 1/5$, $i = 0,\dots,4$ and $U \sim \mathtt{Unif}[0,1]$. Since $\mathbf{e}_1 \preceq \mathbf{e}_3$, we have the following requirements for $\phi(\mathbf{e}_3, \cdot)$: namely $\phi(\mathbf{e}_3, u) \succeq \phi(\mathbf{e}_1, u)$ for $u \in A_i, i = 0,\dots,4$. Thus the function is uniquely determined:

$$\phi(\mathbf{e}_3, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_0 \cup A_1,\\ \mathbf{e}_5 & \text{if} \quad u \in A_2,\\ \mathbf{e}_3 & \text{if} \quad u \in A_3,\\ \mathbf{e}_4 & \text{if} \quad u \in A_4.\end{cases}$$

Also, since $\mathbf{e}_3 \preceq \mathbf{e}_5$, we conclude that

$$\phi(\mathbf{e}_5, u) = \begin{cases} \mathbf{e}_2 & \text{if} \quad u \in A_0 \cup A_1, \\ \mathbf{e}_5 & \text{if} \quad u \in A_2 \cup A_4, \\ \mathbf{e}_3 & \text{if} \quad u \in A_3, \end{cases}$$

Since $\mathbf{e}_1 \preceq \mathbf{e}_4$, we conclude that there are two choices for $\phi(\mathbf{e}_4, u)$. We can have

$$\phi(\mathbf{e}_4, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_0, \\ \mathbf{e}_3 & \text{if} \quad u \in A_1 \cup A_3, \\ \mathbf{e}_2 & \text{if} \quad u \in A_2, \\ \mathbf{e}_4 & \text{if} \quad u \in A_4, \end{cases}$$

but then for $u \in A_1$ we have $\phi(\mathbf{e}_4, u) = \mathbf{e}_3 \not\preceq \mathbf{e}_2 = \phi(\mathbf{e}_5, u)$. We can also have

$$\phi(\mathbf{e}_4, u) = \begin{cases} \mathbf{e}_1 & \text{if} \quad u \in A_1, \\ \mathbf{e}_3 & \text{if} \quad u \in A_0 \cup A_3, \\ \mathbf{e}_2 & \text{if} \quad u \in A_2, \\ \mathbf{e}_4 & \text{if} \quad u \in A_4, \end{cases}$$

but then for $u \in A_0$ we have $\phi(\mathbf{e}_4, u) = \mathbf{e}_3 \not\preceq \mathbf{e}_2 = \phi(\mathbf{e}_5, u)$. Thus, the function is not monotone.