



# Perfect Simulation Of Processes With Long Memory: a 'Coupling Into And From The Past' Algorithm

Aurélien Garivier

► **To cite this version:**

Aurélien Garivier. Perfect Simulation Of Processes With Long Memory: a 'Coupling Into And From The Past' Algorithm. 2011. <hal-00798388v2>

**HAL Id: hal-00798388**

**<https://hal.archives-ouvertes.fr/hal-00798388v2>**

Submitted on 14 Oct 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Perfect Simulation of Processes With Long Memory: A “Coupling Into and From The Past” Algorithm

Aurélien Garivier

October 14, 2013

## Abstract

We describe a new algorithm for the perfect simulation of variable length Markov chains and random systems with perfect connections. This algorithm, which generalizes Propp and Wilson’s simulation scheme, is based on the idea of coupling into and from the past. It improves on existing algorithms by relaxing the conditions on the kernel and by accelerating convergence, even in the simple case of finite order Markov chains. Although chains of variable or infinite order have been widely investigated for decades, their use in applied probability, from information theory to bio-informatics and linguistics, has recently led to considerable renewed interest.

**Keywords:** perfect simulation; context trees; Markov chains of infinite order; coupling from the past (CFTP); coupling into and from the past (CIAFTP)

## 1 Introduction

Since the publication of Propp and Wilson’s seminal paper [21], perfect simulation schemes for stationary Markov chains have been developed and implemented in several fields of applied probabilities, from statistical physics to Bayesian statistics (see for example [18] and references therein, or [12] for an introduction).

In 2002, Comets et al. [5] proposed an extension to processes with long memory: they provided a perfect simulation algorithm for stationary processes called random systems with complete connections [19, 20] or *chains of infinite order* [13]. These processes are characterized by a transition kernel that specifies (given an infinite sequence of past symbols) the probability distribution of the next symbol. Following [16, 17, 2], their work was based on the idea of exploiting the regenerative structures of these processes. The algorithm used by Comets et al. relied on renewal properties that resulted from summable memory decay conditions. As a by-product of the simulation scheme, the authors proved the existence (and uniqueness) of a stationary process  $f$  under suitable hypotheses.

However, these conditions on the kernel seemed quite restrictive and unnecessary. Gallo [9] and Foss et al. [8] showed that different coupling schemes could be designed under alternative assumptions that do not even require kernel continuity. Moreover, the coupling scheme described in [5] strongly relies on

regeneration, not on coalescence. Contrary to Propp and Wilson’s algorithm, this coupling scheme does not converge for all mixing Markov chains; when it does converge, it requires a larger number of steps. Recently, De Santis and Piccioni [7] tried to combine the two algorithms by providing a hybrid method that works with two regimes: coalescence for short memory and regeneration on long scales.

This paper aims to fill the gap between long- and short-scale methods by providing a relatively elaborate, yet elegant coupling procedure that relies solely on coalescence. For a (first order) Markov chain process, this procedure is equivalent to Propp and Wilson’s algorithm. However, our procedure makes it possible to manage more general, infinite memory processes characterized by a continuous transition kernel, as defined in Section 2.

This procedure is based on the idea of exploiting *local* coalescence instead of global loss of memory properties. From an abstract perspective, the algorithm described in Section 3 simply consists in running a Markov chain on an infinite, uncountable set until the first hitting time of a given subset of states. Its concrete implementation involves a dynamical system on a set of labeled trees described in Section 4.

Alternatively, this algorithm may be linked to the algorithm described by Kendall [15], whose adaptation of Propp and Wilson’s idea perfectly simulates the equilibrium distribution of a spacial birth-and-death process to precisely sample from area-interaction point processes. Analogous to the present article, Kendall’s work was based on the idea that if—following the coupled transitions—all possible initial patterns at time  $t < 0$  lead to the same configuration at time 0, then this configuration has the expected distribution: Whenever such a coalescence is observable, perfect simulation is possible. This algorithm was later generalized by Wilson, who named it ‘coupling into and from the past’ (CIAFTP, see [25], Section 7), an appropriate term for the algorithm described in Section 4.

We show that this perfect simulation scheme converges under less restrictive hypotheses than previously required. The paper also provides a detailed description of finite, but large order Markov chains (or variable order Markov chains, see [3]) because they prove very useful in many applications (e.g. information theory [22, 24] or bio-informatics [4]): our algorithm compares favorably with Propp and Wilson’s algorithm on the extended chain in terms of computational complexity; it also compares favorably with the procedure of [5] in terms of convergence speed.

The paper is organized as follows: Section 2 presents the notation and necessary definitions. Section 3 contains the conceptual description of the perfect simulation schemes. An update rule constructed in Section 3.2 served as the key tool. Section 4 contains the detailed description of the algorithm, and Section 5 presents some elements of the complexity analysis. Section 6 demonstrates the relative (compared with other coupling schemes) weakness of the assumptions required for the algorithm to converge. Finally, proofs of technical results are included in the Appendix.

## 2 Notation and definitions

The following section introduces the notation used for algorithms, proofs, and results. Although we primarily relied on standard notation, specific symbols were required, especially concerning trees. This somewhat unusual notation, which is necessary to expose the algorithm as clearly as possible, is central to this paper.

### 2.1 Histories

Corresponding to [5],  $G$  denotes a finite alphabet whose size is denoted by  $|G|$ . For  $k \in \mathbb{N}$ ,  $G^{-k}$  denotes the set of all sequences  $(w_{-k}, \dots, w_{-1})$ , and  $G^* = \cup_{k \geq 0} G^{-k}$ . By convention,  $\varepsilon$  denotes the empty sequence, and  $G^0 = \{\varepsilon\}$ . Referred to as the space of *histories*, the set of  $G$ -valued sequences indexed by the set of negative integers is denoted by  $G^{-\mathbb{N}^+}$ . For  $-\infty \leq a \leq b < 0$  and  $w \in G^{-\mathbb{N}^+}$ , the sequence  $(w_a, \dots, w_b)$  is denoted by  $w_{a:b}$ . An element  $w_{-\infty:-1} \in G^{-\mathbb{N}^+}$  is denoted by  $\underline{w}$ . For  $w \in G^{-k}$ , we write  $|w| = k$ ; for  $\underline{w} \in G^{-\mathbb{N}^+}$ , we write  $|\underline{w}| = \infty$ . For every negative integer  $n$ , we define the projection  $\Pi^n : G^{-\mathbb{N}^+} \rightarrow G^n$  by  $\Pi^n(\underline{w}) = w_{n:-1}$ .

A *trie* is a rooted tree whose edges are labeled with elements of  $G$ . An element  $\underline{w} \in G^{-\mathbb{N}^+}$  can be represented by a path in the infinite, complete trie starting from the root and successively following the edges labeled as  $w_{-1}, w_{-2}, \dots$ . A finite sequence  $s \in G^*$  is represented by an internal node of this infinite trie. Figure 1 illustrates this representation for the binary alphabet  $G = \{0, 1\}$ .

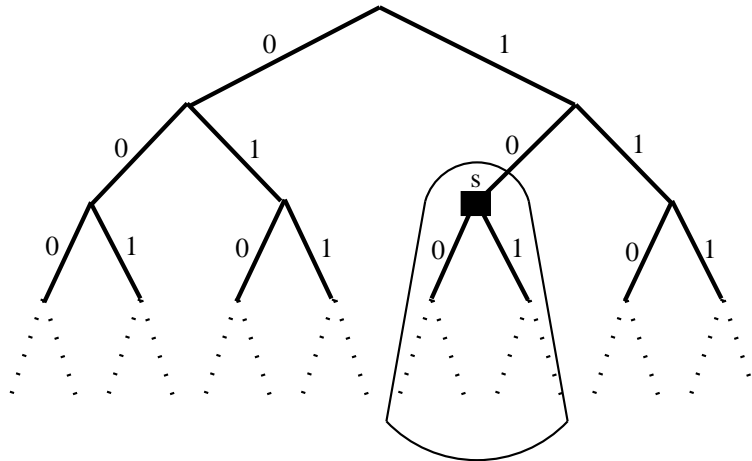


Figure 1: Trie representation of  $G^{-\mathbb{N}^+}$  for the binary alphabet  $G = \{0, 1\}$ . The square represents  $s = (0, 1) \in G^{-2}$ , and  $\mathcal{T}(s)$  is circled. Note that the symbols  $(s_{-2}, s_{-1}) = (0, 1)$  are to be read *from bottom (node  $s$ ) to top (root)*.

### 2.2 Concatenation and suffix

For the two sequences  $w_{a:b}$  and  $z_{c:d}$ , with  $-\infty \leq a \leq b < 0$  and  $-\infty < c \leq d < 0$ , the concatenation of  $w_{a:b}$  and  $z_{c:d}$  is written  $w_{a:b}z_{c:d} = (w_a, \dots, w_b, z_c, \dots, z_d)$ .

In particular, if we take  $a = -\infty$ , then this defines the concatenation  $\underline{z}s$  of a history  $\underline{z}$  and an  $n$ -tuple  $s \in G^{|s|}$ . Note that this notation is different from the convention taken in [5]. If  $a > b$ , then  $w_{a:b}$  is the empty sequence  $\varepsilon$ .

Let  $h \in G^* \cup G^{-\mathbb{N}^+}$ . If  $s \in G^*$  is such that  $|h| \geq |s|$  and  $h_{-|s|:-1} = s$ , we say that  $s$  is a *suffix* of  $h$  and we write  $h \succeq s$ . This defines a partial order  $\succeq$  on  $G^* \cup G^{-\mathbb{N}^+}$ .

### 2.3 Metric

Equipped with the product topology and with the ultra-metric distance  $\delta$  defined by

$$\delta(\underline{w}, \underline{z}) = 2^{\sup\{k < 0 : w_k \neq z_k\}},$$

$G^{-\mathbb{N}^+}$  is a complete and compact set. A ball  $B \subset G^{-\mathbb{N}^+}$  is a set  $\{\underline{z}s : \underline{z} \in G^{-\mathbb{N}^+}\}$  for some  $s \in G^*$ . In reference to the trie representation of  $G^{-\mathbb{N}^+}$ , we write  $s = \mathcal{R}(B)$  for the *root* of  $B$ , and  $\mathcal{T}(s) = B$  for the *tail* of  $s$  (see Figure 1). Note that  $\mathcal{T}(\varepsilon) = G^{-\mathbb{N}^+}$ .

The set of probability distributions on  $G$  is denoted by  $\mathcal{M}(G)$ ; it is endowed with the total variation distance

$$|p - q|_{TV} = \frac{1}{2} \sum_{a \in G} |p(a) - q(a)| = 1 - \sum_{a \in G} p(a) \wedge q(a),$$

where  $x \wedge y$  is the minimum of  $x$  and  $y$ .

### 2.4 Complete suffix dictionaries

A (finite or infinite) set  $D$  of elements of  $G^*$  is called a *complete suffix dictionary* (CSD) if one of the following equivalent properties is satisfied:

- Every sequence  $\underline{w} \in G^{-\mathbb{N}^+}$  has a unique suffix in  $D$ :

$$\forall \underline{w} \in G^{-\mathbb{N}^+}, \exists! s \in D : \underline{w} \succeq s.$$

- $\{\mathcal{T}(s) : s \in D\}$  is a partition of  $G^{-\mathbb{N}^+}$ , in which case we write

$$G^{-\mathbb{N}^+} = \bigsqcup_{s \in D} \mathcal{T}(s).$$

A CSD can be represented by a trie, as illustrated in Fig. 2. This representation suggests that the *depth* of CSD  $D$  is defined as the depth of this trie:

$$d(D) = \sup \{|s| : s \in D\}.$$

Note that  $d(D) = +\infty$  if  $D$  is infinite. The smallest possible CSD is  $\{\varepsilon\}$  (its trie is reduced to the root): it has a depth of 0 and a size of 1. The second smallest is  $G$  with a depth of 1.

If a finite word  $h \in G^*$  has a (unique) suffix in  $D$ , we write  $h \succeq D$ . If  $D$  and  $D'$  are two CSDs such that  $s \succeq D$  as soon as  $s \in D'$ , we write  $D' \succeq D$ . This means that the trie representing  $D'$  entirely covers that of  $D$ , as illustrated in Fig. 2.

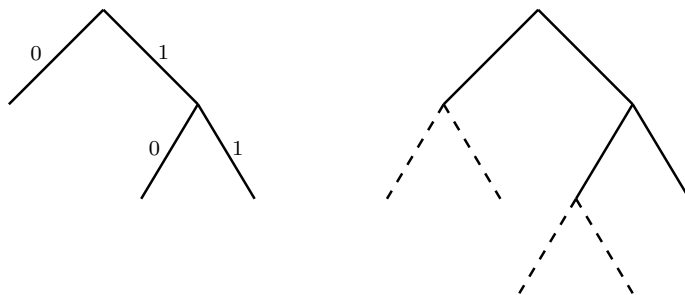


Figure 2: Trie representation of a CSD for the binary alphabet  $G = \{0, 1\}$ . Left: the trie representing the complete suffix dictionary  $D = \{0, 01, 11\}$ . Right:  $\{00, 10, 001, 101, 11\} \succeq \{0, 01, 11\}$ .

## 2.5 Piecewise constant mappings

For a given CSD  $D$ , we say that a mapping  $f$  defined on  $G^{-\mathbb{N}^+}$  is  $D$ -constant if,

$$\forall s \in D, \forall \underline{w}, \underline{z} \in \mathcal{T}(s), f(\underline{w}) = f(\underline{z}) .$$

The mapping  $f$  is constant if and only if it is  $\{\epsilon\}$ -constant, and  $f$  is called *piecewise constant* if there exists a CSD  $D$  such that  $f$  is  $D$ -constant. For every  $h \in G^*$  we define

$$f(h) = f(\mathcal{T}(h)) = \{f(z) : z \in \mathcal{T}(h)\} .$$

Note that, by definition,  $f(h)$  is a set. However, if  $f$  is  $D$ -constant and if  $h \succeq D$ , then  $f(h)$  is a singleton (a set containing exactly one element).

Let  $f$  be a piecewise constant mapping. The set of all CSDs such that  $f$  is  $D$ -constant has a minimal element when ordered by the inclusion relation:  $D^f$  denotes the *minimal CSD* of  $f$ . The minimal CSD  $D^f$  is such that, if  $s \in D^f$ , there exists  $w \in G^*$  such that  $s' = ws_{-|s|+1:-1} \in D^f$  and  $f(s) \neq f(s')$ . If  $f$  is  $D$ -constant, then  $D^f$  can be obtained by recursive pruning of  $D$ , that is by pruning the nodes whose children are leaves with the same  $f$  value (repeating this operation for as long as possible). A  $D$ -constant mapping  $f$  can be represented by the trie  $D$  if each leaf  $s$  of  $D$  is labeled with the common value of  $f(\underline{w})$  for  $w \in \mathcal{T}(s)$ . Figure 3 illustrates the trie representation of a piecewise constant function as well as the pruning operation.

## 2.6 Probability transition kernels

A mapping  $P : G^{-\mathbb{N}^+} \rightarrow \mathcal{M}(G)$  is called a *probability transition kernel*, and we write  $P(\cdot|\underline{w})$  for the image of  $\underline{w} \in G^{-\mathbb{N}^+}$ . We say that  $P$  is *continuous* if it is continuous as an application from  $(G^{-\mathbb{N}^+}, \delta)$  to  $(\mathcal{M}(G), |\cdot|_{TV})$ . For  $s \in G^*$ , we define the *oscillation* of  $P$  on the ball  $\mathcal{T}(s)$  as

$$\eta_P(s) = \sup \left\{ |P(\cdot|\underline{w}) - P(\cdot|\underline{z})|_{TV} : \underline{w}, \underline{z} \in \mathcal{T}(s) \right\} .$$

We say that a process  $(X_t)_{t \in \mathbb{Z}}$  with a distribution  $\nu$  on  $G^{\mathbb{Z}}$  (equipped with the product topology and the product sigma-algebra) is *compatible* with kernel

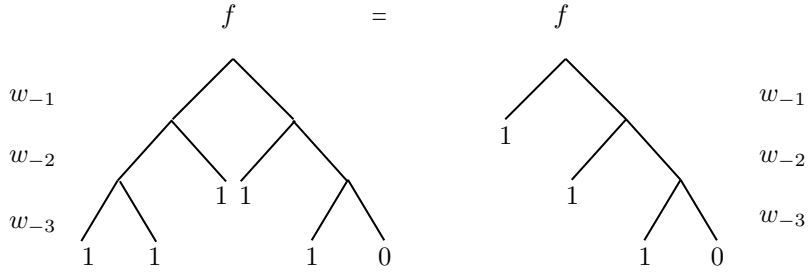


Figure 3: Two representations as labeled tries of the piecewise constant function  $f$  defined for the binary alphabet  $\{0, 1\}^{-\mathbb{N}^+}$  by  $f(\underline{w}) = 0$  if  $\underline{w} \in \mathcal{T}(111)$ , and  $f(\underline{w}) = 1$  otherwise. In the second representation, the trie is minimal: it has been obtained from the first trie by recursively pruning leaves with identical images.

$P$  if the latter is a version of the one-sided conditional probabilities of the former; that is

$$\nu(X_i = g | X_{i+j} = w_j \text{ for all } j \in -\mathbb{N}^+) = P(g | \underline{w})$$

for all  $i \in \mathbb{Z}$ ,  $g \in G$ , and  $\nu$ -almost every  $\underline{w}$ . A classical but key remark states that  $S_t = (\dots, X_{t-1}, X_t)$ ,  $t \in \mathbb{Z}$ , is a homogeneous Markov chain on the compact ultra-metric state space  $G^{-\mathbb{N}^+}$  with a transition kernel  $Q$  given by the relation,

$$\forall \underline{w}, \underline{z} \in G^{-\mathbb{N}^+}, \quad Q(\underline{z} | \underline{w}) = P(z_{-1} | \underline{w}) 1_{\bigcap_{i < 0} \{z_{i-1} = w_i\}}.$$

## 2.7 Update rules

An application  $\phi : [0, 1[ \times G^{-\mathbb{N}^+} \rightarrow G$  is called an *update rule* for a kernel  $P$  if, for all  $\underline{w} \in G^{-\mathbb{N}^+}$  and for all  $g \in G$ , the Lebesgue measure of  $\{u \in [0, 1[ : \phi(u, \underline{w}) = g\}$  is equal to  $P(g | \underline{w})$ . In other words, if  $U$  is a random variable uniformly distributed on  $[0, 1[$ , then  $\phi(U, \underline{w})$  has a distribution  $P(\cdot | \underline{w})$  for all  $\underline{w} \in G^{-\mathbb{N}^+}$ . For any continuous kernel  $P$ , Section 3.2 details the construction of an update rule  $\phi_P$  such that,

$$\text{for all } s \in G^*, \quad 0 \leq u < 1 - |G|\eta_P(s) \implies \phi_P(u, \cdot) \text{ is constant on } \mathcal{T}(s). \quad (1)$$

The following lemma (proved in the Appendix) states the basic observation that permits us to design an algorithm working in finite time, which is applicable even for kernels that are not piecewise continuous.

**Lemma 1.** *For all  $u \in [0, 1[$ , the mapping  $\underline{w} \rightarrow \phi_P(u, \underline{w})$  is continuous, i.e. piecewise constant.*

## 3 Abstract description of the perfect simulation scheme

Given a continuous transition kernel  $P$ , two questions arise:

1. Does a stationary distribution  $\nu$  compatible with  $P$  exist? If it exists, is it unique?
2. If  $\nu$  exists, how can we sample finite trajectories from that distribution?

Several authors [5, 7, 9] have contributed to answering these questions in the past decade. Their approach consisted in showing that there exists a simulation scheme that draws samples of  $\nu$ . This algorithm was based on the idea of coupling from the past. In accordance with these authors, we addressed these questions by constructing a new perfect simulation scheme that requires looser conditions on the kernel and that converges faster than existing algorithms. The following section describes the general principle of this algorithm. Practical details concerning its implementation are given in Section 4.

### 3.1 Perfect simulation by coupling into the past

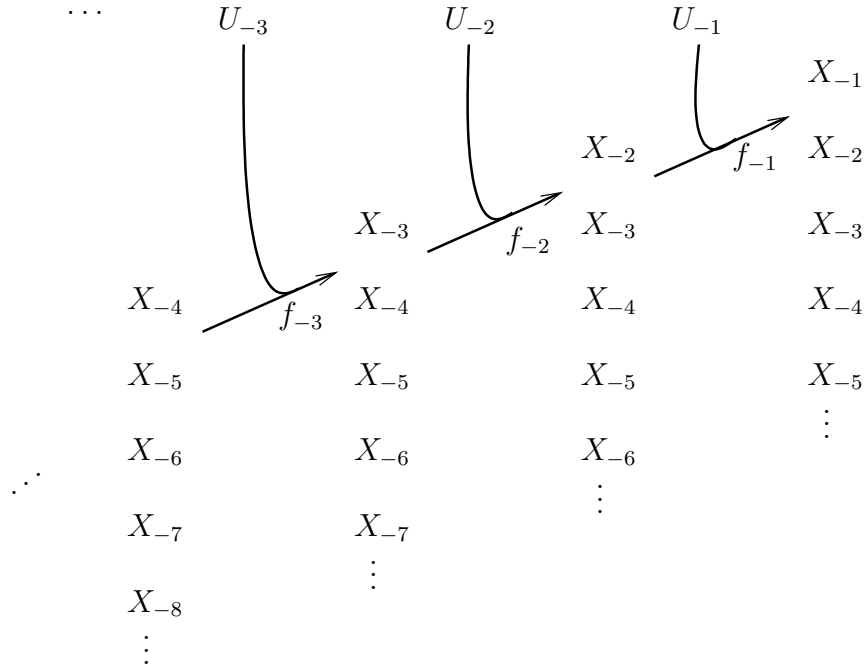


Figure 4: Perfect simulation scheme.

Let  $n$  be a negative integer. In order to draw  $(X_n, \dots, X_{-1})$  from a stationary distribution compatible with  $P$ , we use a semi-infinite sequence of independent random variables  $(U_t)_{t < 0}$  defined on a probability space  $(\Omega, \mathcal{A}, P)$  and uniformly distributed on  $[0, 1]$ . The variable  $X_t$  is deduced from  $U_t$  and from the past symbols  $X_{t-1}, X_{t-2}, \dots$ , as depicted in Fig. 4. Those past symbols are unknown, but the regularity of  $P$  makes it nevertheless possible to sometimes compute  $X_t$ .

For each  $t < 0$ , let  $f_t$  be the random function  $G^{-\mathbb{N}^+} \rightarrow G^{-\mathbb{N}^+}$  defined by



$f_t(\underline{w}) = \underline{w}\phi_P(U_t, \underline{w})$ .<sup>1</sup> Beware of the index shift: if  $\underline{z} = f_t(\underline{w})$ , then  $z_{-1} = \phi_P(U_t, \underline{w})$  and  $z_i = w_{i+1}$  for  $i < -1$ .

In addition, let  $F_t = f_{-1} \circ \dots \circ f_t$  and, for any negative integer  $n$ ,  $H_t^n = \Pi^n \circ F_t$ . Proposition 1 below shows that the continuity of  $P$  implies that  $H_t^n$  is piecewise constant. We define

$$\tau(n) = \sup\{t < 0 : H_t^n \text{ is constant}\},$$

where, by convention,  $\tau(n) = -\infty$  if  $H_t^n$  is not constant for all  $t < -1$ . When  $\tau(n)$  is finite, the result of the procedure is the image  $\{X_{n:-1}\}$  of the constant mapping  $H_{\tau(n)}^n$ . We can easily verify that  $X_{n:-1}$  has the expected distribution (see [21, 5]).

**Remark 1.** For  $t > n$ ,  $H_t^n$  cannot be constant because it holds that  $(H_t^n(\underline{w}))_k = w_k$  for all  $n \leq k < t$ . Thus,  $\tau(n) = \sup\{t \leq n : H_t^n \text{ is constant}\} \leq n$ .

Observe also that the sequence  $(\tau(n))_n$  is a non-increasing sequence of stopping times with respect to the filtration  $(\mathcal{F}_s)_s$ , where  $\mathcal{F}_s = \sigma(U_t : t \geq s)$  when  $s$  decreases.

From a theoretical perspective, this CIAFTP algorithm simply consists in running an instrumental Markov chain until a given hitting time. In fact, the recursive definition given above shows that the sequence  $(H_t^n)_{t \leq 1}$  is a homogeneous Markov chain on the set of functions  $G^{-\mathbb{N}^+} \rightarrow G^n$ . The algorithm terminates when this Markov chain hits the set of constant mappings. Such a procedure seems to be purely abstract because it involves infinite, uncountable objects. However, Section 4 shows how this Markov chain on the set of functions  $G^{-\mathbb{N}^+} \rightarrow G^n$  can be handled with a finite memory. Before we provide the detailed implementation of the algorithm, we first present the construction of the update rule and the sufficient conditions for the finiteness of the stopping time  $\tau(n)$  in Section 3.2.

### 3.2 Constructing the update rule $\phi_P$

The algorithm that is abstractly depicted above and detailed in Section 4 crucially relies on the update rule  $\phi_P$  that satisfies Equation (1). We present here the construction of this update rule for a given continuous kernel  $P$ . In short, for each  $k$ -tuple  $z \in G^{-k}$ , the construction of  $\phi_P$  relies on a coupling of the conditional distributions  $\{P(\cdot | \underline{z}) : \underline{z} \in \mathcal{T}(z)\}$ . The simultaneous construction of all of these couplings requires a few definitions and properties, which we state in this section and prove in the Appendix.

Provide  $G$  with any order  $<$ , so that  $G^{-\mathbb{N}^+}$  can be equipped with the corresponding lexicographic order :  $\underline{w} < \underline{z}$  if there exists  $k \in -\mathbb{N}$  such that, for all  $j > k$ ,  $w_j = z_j$ , and  $w_k < z_k$ . The continuity of  $P$  is locally quantified by some coupling factors, which we define along with the coefficients that

<sup>1</sup>Regarding measurability issues: if the set of functions  $G^{-\mathbb{N}^+} \rightarrow G^{-\mathbb{N}^+}$  is equipped with the topology induced by the distance  $\underline{\delta}$  defined by

$$\underline{\delta}(f_1, f_2) = \sum_{w \in G^*} (2|G|)^{-|w|} \sup \left\{ \delta(f_1(\underline{z}_1), f_2(\underline{z}_2)) : \underline{z}_1, \underline{z}_2 \in \mathcal{T}(w) \right\}$$

and with the corresponding Borel sigma-algebra, then the measurability of  $f_t$  follows from Lemma 1.

are necessary for the construction of the update rule  $\phi_P$ . For all  $g \in G$ , let  $A_{-1}(\varepsilon) = a_{-1}(g|\varepsilon) = 0$ . For all  $k \in \mathbb{N}$  and all  $z \in G^{-k}$ , let

$$\begin{aligned} a_k(g|z_{-k:-1}) &= \inf \{P(g|\underline{w}) : \underline{w} \in \mathcal{T}(z_{-k:-1})\}, \\ A_k(z_{-k:-1}) &= \sum_{g \in G} a_k(g|z_{-k:-1}), \\ A_k^- &= \inf_{s \in G^{-k}} A_k(s), \\ \alpha_k(g|z_{-k:-1}) &= A_{k-1}(z_{-k+1:-1}) + \sum_{h < g} \{a_k(h|z_{-k:-1}) - a_{k-1}(h|z_{-k+1:-1})\}, \end{aligned} \tag{2}$$

$$\beta_k(g|z_{-k:-1}) = A_{k-1}(z_{-k+1:-1}) + \sum_{h \leq g} \{a_k(h|z_{-k:-1}) - a_{k-1}(h|z_{-k+1:-1})\}. \tag{3}$$

Note that, with our conventions,  $a_0(g|\varepsilon) = \inf\{P(g|\underline{z}) : \underline{z} \in G^{-\mathbb{N}^+}\}$ . Moreover, if  $s$  and  $s' \in G^*$  are such that  $s \succeq s'$ , then for all  $g \in G$  it holds that  $a_k(g|s) \geq a_k(g|s')$ ,  $A_k(s) \geq A_k(s')$ , and the sequence  $(A_k^-)_k$  is non-decreasing.

The following propositions gather some elementary ideas that are used in the following; they are proved in the Appendix.

**Proposition 1.** *The coupling factors of the kernel  $P$  satisfy the following inequalities: for all  $s \in G^*$ ,*

$$1 - (|G| - 1)\eta_P(s) \leq A_{|s|}(s) \leq 1 - \eta_P(s). \tag{4}$$

**Proposition 2.** *The following assertions are equivalent:*

- (i) *the kernel  $P$  is continuous;*
- (ii) *for every  $\underline{w} \in G^{-\mathbb{N}^+}$ ,  $\eta_P(w_{-k:-1})$  tends to 0 when  $k$  goes to infinity;*
- (iii) *when  $k$  goes to infinity,*

$$\sup \{\eta_P(s) : s \in G^{-k}\} \rightarrow 0;$$

- (iv) *for all  $\underline{w} \in G^{-\mathbb{N}^+}$ ,  $A_k(w_{-k:-1}) \rightarrow 1$  when  $k \rightarrow \infty$ ;*

- (v)  *$A_k^- \rightarrow 1$  when  $k$  goes to infinity.*

**Proposition 3.** *Let  $P$  be a continuous kernel, and let  $\alpha_k(\cdot|\cdot)$  and  $\beta_k(\cdot|\cdot)$  be defined as in (2) and (3). Then, for every  $\underline{w} \in G^{-\mathbb{N}^+}$ ,*

$$[0, 1[ = \bigsqcup_{g \in G, k \in \mathbb{N}} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[.$$

*In other words, for every  $u \in [0, 1[$  and every  $\underline{w} \in G^{-\mathbb{N}^+}$ , there exists a unique  $k \in \mathbb{N}$  and a unique  $g \in G$  such that  $u \in [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[$ .*

Figure 5 illustrates Proposition 3 on a three-symbol alphabet. Thanks to Proposition 3, we can now define the following update rule and verify that it satisfies Equation (1).

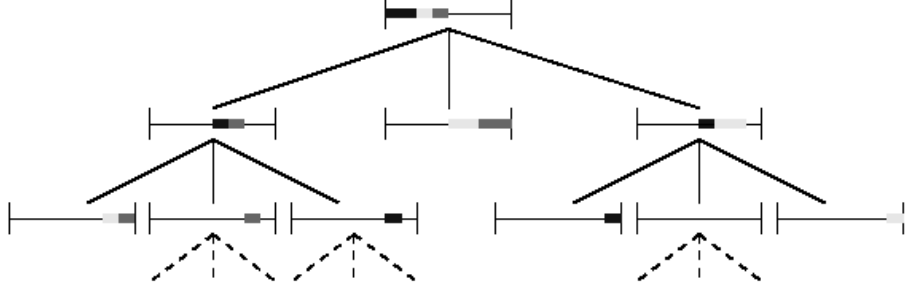


Figure 5: Graphical representation of an update rule  $\phi_P$  on the alphabet  $\{0, 1, 2\}$ : for each  $w_{-k:-1}$ , the intervals  $[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[$  are represented in black ( $g = 0$ ), light grey ( $g = 1$ ), and medium grey ( $g = 2$ ). For example,  $P(1|1) = \alpha_0(1|\varepsilon) + \alpha_1(1|1) = 1/8 + 1/4$ , and  $P(0|00) = \alpha_0(0|\varepsilon) + \alpha_1(0|0) + \alpha_2(0|00) = 1/4 + 1/8 + 0$ .

**Definition 1.** Let  $\phi_P : [0, 1[ \times G^{-\mathbb{N}^+} \rightarrow G$  be defined as

$$\phi_P(u, \underline{w}) = \sum_{g \in G, k \in \mathbb{N}} g \mathbb{1}_{[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[}(u).$$

In words, for every  $u \in [0, 1[$  and for every  $\underline{w} \in G^{-\mathbb{N}^+}$ ,  $\phi_P(u, \underline{w})$  is the unique symbol  $g \in G$  such that there exists  $k \in \mathbb{N}$  satisfying

$$u \in [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[.$$

**Proposition 4.** The mapping  $\phi_P$  of Definition 1 is an update rule satisfying Equation (1):

$$\forall s \in G^*, \forall u \in [0, 1], \forall \underline{w}, \underline{z} \in \mathcal{T}(s), \quad u < A_{|s|}(s) \implies \phi_P(u, \underline{w}) = \phi_P(u, \underline{z}).$$

As a consequence, for every  $s \in G^*$  and every  $u < A_{|s|}(s)$ , we can define  $\phi_P(u, s)$  as the value common to  $\phi_P(u, \underline{w})$  for all  $\underline{w} \in \mathcal{T}(s)$ .

### 3.3 Convergence

Sufficient conditions on  $P$  are given in [5, 7] to ensure that  $\tau(n)$  is almost surely finite (or even that  $\tau(n)$  has bounded expectation). In addition, the authors prove that the almost-sure finiteness of  $\tau(n)$  is a sufficient condition to prove the existence and uniqueness of a stationary distribution  $\nu$  compatible with  $P$  (see [5], Theorem 4.3 and corollaries 4.12 and 4.13). As a by-product, they obtained a simulation algorithm for sample paths of  $\nu$ : if  $U = (U_t)_{t \in \mathbb{Z}}$  is a sequence of independent, uniformly distributed random variables, then  $\Phi : [0, 1]^{\mathbb{Z}} \rightarrow G^{\mathbb{Z}}$  can be defined such that  $\Phi(U)_t = \phi_P(U_{t-1}, \Phi(U)_{-\infty:t-1})$  for all  $t$ , and

$$\nu = \mathbb{P}(\Phi(U) \in \cdot),$$

the law of  $\Phi(U)$ , is stationary and compatible with  $P$ .

However, in [5], the authors impose fairly restrictive conditions on  $P$ : they require that

$$\sum_{m \geq 0} \prod_{k=0}^m A_k^- = \infty,$$

and, in particular, that the chain satisfies the Harris condition

$$A_0^- = \sum_{g \in G} \inf \left\{ P(g | \underline{z}) : \underline{z} \in G^{-\mathbb{N}^+} \right\} > 0.$$

By using a Kalikow-type decomposition of the kernel  $P$  as a mixture of Markov chains of all orders, the authors prove that the process regenerates and that the stopping time

$$\tau'(n) = \sup \{ t \leq n : H_t^t \text{ is constant} \}$$

is almost surely finite under these conditions.

This condition is clearly sufficient but certainly not necessary for  $\tau(n)$  to be finite. Consider, for example, a first-order Markov chain: Although Propp and Wilson [21] have shown that the stopping time  $\tau(n)$  of the optimal update rule is almost surely finite for every mixing chain (and, under some conditions, that  $\tau(n)$  has the same order of magnitude as the mixing time of the chain),  $\tau'(n)$  is almost surely infinite as soon as the Dobrushin coefficient  $A_0^-$  of the chain is 0. In this paper, we close the gap by providing a Propp–Wilson procedure for general continuous kernels that may converge even if the process is not regenerating. For first-order Markov chains,  $\phi_P(u, \underline{w})$  depends only on  $w_{-1}$ , and the algorithm presented in this paper corresponds to Propp and Wilson’s exact sampling procedure.

Since the publication of [5], these results have been generalized [9, 7], which included relaxing the conditions on the kernel and proposing other particular conditions for different cases. Gallo [9] showed that the kernel  $P$  need not be continuous to ensure the existence of  $\nu$ , nor to ensure the finiteness of  $\tau(n)$ : he gives an example of a non-continuous regenerating chain (see also the final remark of Section 6). De Santis and Piccioni [7] have proposed another algorithm, which combines the ideas of [5] and [21]: they propose a hybrid simulation scheme that works with a Markov regime and a long-memory regime. We take a different, more general approach. Our procedure generalizes the sampling schemes of [5] and [21] in a single, unified framework.

## 4 The coupling into and from the past algorithm

This section gives a detailed description of the algorithm that permits us to effectively compute the mappings  $H_t^n$ . The problem is that the mapping domain is the infinite space  $G^{-\mathbb{N}^+}$ , so that no naive implementation is possible. We are able to solve this problem because, for each  $t$ , the mapping  $H_t^n$  is piecewise continuous and thus can be represented by a random but finite object: namely, by its trie representation defined in Section 2.5.

### 4.1 Description of the algorithm

Consider a continuous kernel  $P$  and its update rule  $\phi_P$  given by Definition 1. For each  $u \in [0, 1]$ , Proposition 1 shows that the mapping  $\phi_P(u, \cdot)$  is piecewise con-

stant; its minimal CSD is written  $D(u) = D^{\phi_P(u, \cdot)}$ . Algorithm 1 shows how the mappings  $H_t^n$  (defined in Section 3) can be constructed recursively using only finite memory. For simplicity, this algorithm is presented as a pseudo-code involving mathematical operations and ignoring specific data structures. It should, however, be easy to deduce a real implementation from this pseudo-code. A matlab implementation is available online at <http://www.math.univ-toulouse.fr/~agarivie/Telecom/context/>. It contains a demonstration script illustrating the perfect simulation of the processes mentioned in Sections 5.2 and 6.

---

**Algorithm 1:** Coupling from and into the past for continuous kernels.

---

**Input:** update rule  $\phi_P$ , size  $-n$  of the path to sample

```

1  $t \leftarrow 0$ ;
2  $D_t^n \leftarrow G^n$ ;
3 for all  $s \in G^n, H_t^n(s) \leftarrow \{s\}$ ;
4 while  $|D_t^n| > 1$  do
5    $t \leftarrow t - 1$ ;
6   draw  $U_t \sim \mathcal{U}([0, 1])$  independently;
7    $D(U_t) \leftarrow$  the minimal trie of  $U_t$ ;
8   foreach  $s \in D(U_t)$  do
9      $\{g_t[s]\} \leftarrow \phi_P(U_t, s)$ ;
10    if  $sg_t[s] \succeq D_{t+1}^n$  then
11       $E_t^n[s] \leftarrow \{s\}$ ;
12    else
13       $E_t^n[s] \leftarrow \{h \in G^* : hg_t[s] \in D_{t+1}^n(sg_t[s])\}$ ;
14   $E_t^n \leftarrow \bigcup_{s \in D(U_t)} E_t^n[s]$ ;
15  Claim 1:  $E_t^n$  is a CSD;
16  Claim 2:  $H_t^n$  is  $E_t^n$ -constant, and, for all  $s \in E_t^n$ ,
17   $H_t^n(s) = H_{t+1}^n(sg_t[s])$  is a singleton;
18   $D_t^n \leftarrow$  the minimal CSD of  $H_t^n$  obtained by pruning  $E_t^n$ 

```

**Output:**  $X_{n:-1}$  such that, for all  $\underline{z} \in G^{-\mathbb{N}^+}$ ,  $H_t^n(\underline{z}) = \{X_{n:-1}\}$

---

For every  $t < 0$ , the mapping  $H_t^n$  being piecewise constant, we can define  $D_t^n = D^{H_t^n}$ . Note that the definition of  $H_0^n$  in the initialization step is consistent with the general definition  $H_t^n = \Pi^n \circ F_t$  because the natural definition of  $F_0$  is the identity map on  $G^{-\mathbb{N}^+}$ . Algorithm 1 successively computes  $H_{-1}^n, H_{-2}^n, \dots$  and stops for the first  $t \leq n$  such that  $H_t^n$  is constant.

The key step is the derivation of  $H_t^n$  and  $D_t^n$  from  $H_{t+1}^n, D_{t+1}^n$ , and  $U_t$ . This derivation is illustrated in Figure 6 and consists of three steps:

**STEP 1:** Compute the minimal trie  $D(U_t)$  of  $\phi(U_t, \cdot)$ .

**STEP 2:** Compute the trie  $E_t^n$  such that  $H_t^n$  is  $E_t^n$ -constant by completing  $D(U_t)$  with portions of  $D_{t+1}^n$ . Namely, for every  $s \in D(U_t)$ , there are two cases:

- Either  $sg_t[s] \succeq D_{t+1}^n$ , then knowing that  $(X_{t-|s|}, \dots, X_{t-1}) = s$  together with  $U_t$  and  $H_t^n$  is sufficient to determine  $X_{n:-1}$  (see the dashed lines in Figure 6),
- or some additional symbols in the past are required by  $H_{t+1}^n$ , and a subtree of  $D_{t+1}^n$  has to be inserted instead of  $s$  (see the dotted circled subtree in Figure 6).

**STEP 3:** Prune  $E_t^n$  to obtain the minimal trie  $D_t^n$  of  $H_t^n$ .

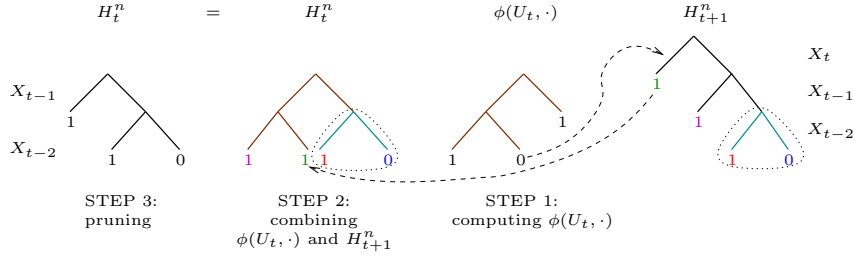


Figure 6: Three-step process for the trie representation of  $H_t^n$  deduced from the representations of  $H_{t+1}^n$  and  $U_t$ . Here,  $G = \{0, 1\}$  and  $n = 1$ . Recall that  $\phi(U_t, \cdot)$  gives  $X_t$  according to  $(X_{t-1}, X_{t-2}, \dots)$ , and that  $H_{t+1}^n$  gives  $X_{-1}$  according to  $(X_t, X_{t-1}, X_{t-2}, \dots)$ . In Step 2, dashed lines illustrate the first case (if  $(X_{t-2}, X_{t-1}) = (0, 1)$ , then  $X_t = \phi(U_t, \dots 01) = 0$  and  $X_{-1} = H_{t+1}^n(\dots 010) = H_{t+1}^n(\dots 0) = 1$ ); for the second case, a dotted line circles the subtree of  $D_{t+1}^n$  to be inserted into  $E_t^n$ .

From a mathematical perspective, Algorithm 1 can be considered a run of an instrumental, homogeneous Markov chain on the set of  $|G|^n$ -ary trees whose leaves are labeled as  $G^n$ , which is stopped as soon as a tree of depth 0 is reached. Figure 6 illustrates one iteration of this chain, corresponding to one loop of the algorithm.

Algorithm 1 thus closely resembles the high-level method termed ‘coupling into and from the past’ in [25] (see Section 7, in particular Figure 7). In fact, in addition to coupling the trajectories starting *from* all possible states at past time  $t$ , we use a coupling of the conditional distributions before time  $t$  (that is, into the past). Our algorithm is slightly different because we want to sample  $X_{n:-1}$  in addition to  $X_{-1}$ . The CSD  $D_t^n$  corresponds to the state denoted by  $X$  in [25], and  $H_t^n$  corresponds to  $F$ .

## 4.2 Correctness of Algorithm 1

To prove the correctness of Algorithm 1, that is, the correctness of the update rule deriving  $H_t^n$  from  $H_{t+1}^n$ , we must verify the two claims on lines 15 and 16.

**Claim 1:**  $E_t^n$  is a CSD.

Every  $h \in E_t^n[s]$  is such that  $h \succeq s$ . Let  $\underline{w} \in \mathcal{T}(s)$ , and let  $\{hg_t[s]\} = D_{t+1}^n(\underline{w}g_t[s])$ . Then  $h \in E_t^n[s]$  and  $\underline{w} \in \mathcal{T}(h)$ , so that

$$\mathcal{T}(s) = \bigsqcup_{h \in E_t^n[s]} \mathcal{T}(h).$$

Because  $G^{-\mathbb{N}^+} = \sqcup_{s \in D(U_t)} \mathcal{T}(s)$ , the result follows.

**Claim 2:**  $H_t^n$  is  $E_t^n$ -constant, and  $H_t^n(s) = H_{t+1}^n(sg_t[s])$  is a singleton for all  $s \in E_t^n$ .

We prove that  $H_t^n$  is  $E_t^n$ -constant by induction on  $t$ , and the formula for  $H_t^n(s)$  is a by-product of the proof. For  $t = 0$ , this is obvious if we write  $E_0^n = D_0^n = G^n$ . For  $t < 0$ , let  $h \in E_t^n$ . By construction,  $h \succeq D(U_t)$ : denote by  $s$  the suffix of  $h$  in  $D(U_t)$ . Then  $\phi_P(U_t, h)$  is the singleton  $\{g_t[s]\}$ . By construction,  $hg_t[s] \succeq D_{t+1}^n$ , and therefore  $H_t^n(h) = H_{t+1}^n(hg_t[s])$  is a singleton by the induction hypothesis.

### 4.3 Computational complexity

For a given kernel, the random number of elementary operations performed by a computer during a run of Algorithm 1 is a complicated variable to analyze because it depends not only on the number  $\tau(n)$  of iterations, but also on the size of the trees  $D_t^n$  involved. Moreover, the basic operations on trees (traversal, lookup, node insertion or deletion, etc.) performed at each step have a computational complexity that depends on the implementation of the trees.

As a first approximation, however, we can consider the cost of these operations to be a unit. Then, the computational complexity of the algorithm is bounded by the number of such operations in a run. A brief inspection of Algorithm 1 reveals that the complexity of a run is proportional to the sum of the number of nodes of  $D_t^n$  for  $t$  from  $\tau(n)$  to  $-1$ . Taking into account the complexity of the basic tree operations, this would typically lead to a complexity  $O\left(\sum_{t=\tau(n)}^{-1} |D_t^n| \log |D_t^n|\right)$ .

Thus, an analysis of the computational complexity of Algorithm 1 requires simultaneous bounding of the number of iterations  $\tau(n)$  and the size of the trees  $D_t^n$ . For a general kernel  $P$ , this involves not only the mixing properties of the corresponding process, but also the oscillation of the kernel itself, which amounts to a very challenging task that surpasses the scope of this paper. Nevertheless, Section 5 contains some analytical elements. The problematic issues are considered successively: First we give a crude bound on  $\tau(n)$ , then we prove a bound on the size of  $D_t^n$ 's for finite memory processes.

## 5 Bounding the size of $D_t^n$

We establish sufficient conditions for the algorithm to terminate and define the bound on the expectation of the depth of  $D_t^n$ . We then focus on the special, yet important, case of (finite) variable length Markov chains.

### 5.1 Almost sure termination of the coupling scheme

In general, the size of the CSD  $D_t^n$  may be arbitrary with a positive probability. The conditions that ensure the finiteness of  $\tau(n)$ , defined above, from which the bounds on  $D_t^n$  can be deduced are given in [5]. However, these conditions are quite restrictive: in particular, they require that  $A_0(\varepsilon) > 0$ . The hybrid simulation scheme used in [7], which allows for  $A_0(\varepsilon) = 0$ , somewhat relaxes these conditions.

We define a crude bound, ignoring the coalescence possibilities of the algorithm: the depth of the current tree at time  $t$  is defined as  $L_t^n = d(D_t^n)$ . Thereby an immediate inspection of Algorithm 1 yields

$$\begin{cases} L_t^t \leq \max\{X_t, L_{t+1}^{t+1} - 1, 1\} & \text{if } t < n, \text{ and} \\ L_t^n \leq \max\{X_t, L_{t+1}^n - 1\} & \text{if } t \geq n, \end{cases}$$

where  $X_t = d(D(U_t))$  represents i.i.d. random variables such that, for all  $k \in \mathbb{N}$ ,  $\mathbb{P}(X_t \leq k) = A_k^-$ . Therefore,

$$\mathbb{P}(L_t^n \leq k) \geq \mathbb{P}(L_{t+1}^n \leq k+1)A_k^- \geq \prod_{j=k}^{k-t-1} A_j^-.$$

As previously demonstrated in [5], it results that  $\tau(n)$  is almost surely finite as soon as

$$\sum_{m \geq 0} \prod_{k=0}^m A_k^- = \infty.$$

In addition, it follows that

$$\mathbb{E}[L_t^n] \leq \sum_{k=1}^n \left( 1 - \prod_{j=k}^{\infty} A_j^- \right).$$

## 5.2 Finite context trees

It is easy to upper-bound the size of  $D_t^n$  independently of  $t \leq n$  for at least one case: when the kernel  $P$  actually defines a *finite context tree*, that is, when the mapping  $\underline{w} \rightarrow P(\cdot|\underline{w})$  is piecewise constant. In other words,  $P(\cdot|s)$  is a singleton for each  $s \in D$ , where  $D$  denotes the minimal CSD of this mapping.

Yet, the simulation scheme described above is useful even in that case: Although the “plain” Propp-Wilson algorithm could be applied to the first-order Markov chain  $(X_{t+1:t+d(D)})_{t \in \mathbb{Z}}$  on the extended state space  $G^{d(D)}$ , the computational complexity of such an algorithm might become rapidly intractable if the depth  $d(D)$  is large. In contrast, the following property shows that our algorithm retains a possibly much more limited complexity. It is precisely because of these qualities of *parsimony* that finite context trees have proved successful in many applications, from information theory and universal coding (see [22, 24, 6, 11]) to biology ([1, 4]) and linguistics [10].

We say that a CSD  $D$  is *prefix-closed* if every prefix of any sequence in  $D$  is the suffix of an element of  $D$ :

$$\forall s \in D, \forall k \leq |s|, \exists w \in D : w \succeq s_{-|s|:-k}.$$

A prefix-closed CSD satisfies the following property:

**Lemma 2.** *If  $D$  is a prefix-closed CSD, then  $ha \succeq D$  for all  $h \in D$  (or, equivalently, for all  $h \succeq D$ ) and for all  $a \in G$ .*

**Proof:** If  $h \in G^*$  is such that  $ha \succeq D$  does not hold for some  $a \in G$ , then (because  $D$  is a CSD) there exists  $s \in D$  and  $s' \in G^* \setminus \{\epsilon\}$  such that  $s = s'ha$ .



But then  $s'h$  is a prefix of  $s$  and, by the prefix-closure property, there exists  $w \in D$  such that  $w \succeq s'h$ . Thus, we cannot have  $h \succeq D$ .

We define the *prefix closure*  $\overleftarrow{D}$  of a CSD  $D$  as the minimal prefix-closed set containing  $D$ , that is, the set of maximal elements (for the partial order  $\succeq$ ) of

$$\tilde{D} = \{s_{-|s|:-k} : s \in D, k \leq |s|\} .$$

In other words,  $\overleftarrow{D}$  is the smallest set such that, for all  $w \in \tilde{D}$ , there exists  $s \in \overleftarrow{D}$  such that  $s \succeq w$ .

Clearly,  $|\overleftarrow{D}| \leq |\tilde{D}| \leq |D| \times d(D)$ . This bound is generally pessimistic: many CSDs are already prefix-closed, and for most CSDs,  $|\overleftarrow{D}|$  is of the same order of magnitude as  $|D|$ . But, in fact, for each positive integer  $n$ , we can show that there exists a CSD  $D$  of size  $n$  such that  $|\overleftarrow{D}| \geq c|D|^2$  for some constant  $c \approx 0.4$ .

Now, assume that  $D \neq \{\epsilon\}$ , i.e. that  $P$  is not memoryless.

**Proposition 5.** *For each  $t \leq n$  and for all  $k < t$ ,  $\overleftarrow{D} \succeq D_t^n$ . Therefore,*

$$|D_t^n| \leq |\overleftarrow{D}| \leq |D| \times d(D) .$$

**Proof:** We show that  $\overleftarrow{D} \succeq D_t^n$  by induction on  $t$ . First, as  $P$  is not memoryless,  $\overleftarrow{D} \succeq D_{-1}^{-1} = G$ . Second, assume that  $\overleftarrow{D} \succeq D_{t+1}^n$ : it is sufficient to prove that  $H_t^n$  (or  $H_t^t$ , if  $t \geq n$ ) is  $\overleftarrow{D}$ -constant. Observing that  $\overleftarrow{D} \succeq D$ , for every  $U_t \in [0, 1[$  and for every  $s \in \overleftarrow{D}$ , it holds that  $\phi_P(U_t, s)$  is a singleton  $\{g_t[s]\}$ . By successively using the lemma and the induction hypothesis, we have  $sg_t[s] \succeq \overleftarrow{D} \succeq D_t^n$ , thus  $H_t^n(s) = H_{t+1}^n(sg_t[s])$  is also a singleton. Finally, for  $t = n$ ,  $H_t^t$  is  $\overleftarrow{D}$ -constant because  $\overleftarrow{D} \neq \{\epsilon\}$ .

It has to be emphasized that Proposition 5 provides only an upper-bound on the size of  $D_t^n$ : in practice,  $D_t^n$  is often observed to be much smaller. Even for non-sparse, large order Markov chains of an order of  $d$ , it is possible for Algorithm 1 to be faster than the Propp-Wilson algorithm on  $G^d$ , which generally requires the consideration of  $|G|^d$  states at each iteration. Interested readers may want to run the matlab experiments available at <http://www.math.univ-toulouse.fr/~agarivie/Telecom/context/>.

## 6 Example: A continuous process with long memory

This section briefly illustrates the strengths of Algorithm 1 in comparison with the other existing CFTP algorithms for infinite memory processes. We focus on a process that cannot be simulated by other methods, although, of course, Algorithm 1 is also relevant for all processes mentioned in [5, 7], which we refer to for further examples.

The example we consider involves a non-regenerating kernel on the binary alphabet  $G = \{0, 1\}$ . It is such that  $a_0 = 0$  and that the convergence of the coupling coefficients is slow, so that neither the perfect simulation scheme of [5], nor its improvement by [7] can be applied. Yet, a probabilistic upper-bound on

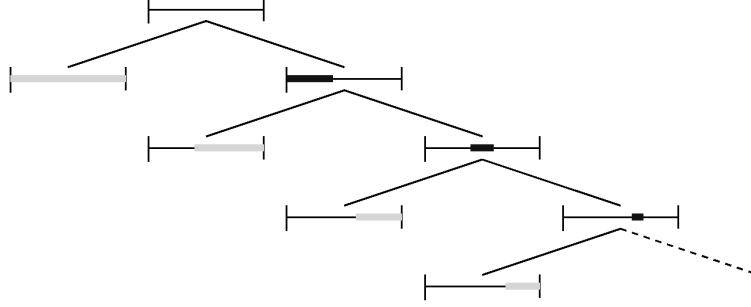


Figure 7: Graphical representation of the update rule for the kernel defined in (5). Dark grey corresponds to 0. Light grey corresponds to 1.

the stopping time  $\tau$  of Algorithm 1 can be given, which proves that there exists a compatible stationary process. For all  $k \geq 0$ , let

$$P(0|01^k) = 1 - 1/\sqrt{k}. \quad (5)$$

Figure 7 shows the coupling coefficients of  $P$ . As  $P(1|0) = \lim_{k \rightarrow \infty} P(0|01^k) = 1$ ,  $a_0 = 0$ . Moreover, for  $k \geq 0$ , it holds that  $A_{k+1} = A_k(01^k) = 1 - 1/\sqrt{k}$ , so that

$$\sum_n \prod_{k=2}^n A_k^- < \infty$$

and the continuity conditions of [5, 7] do not apply.

We demonstrate that the algorithm described above can be used to simulate samples of a process  $X$  with the specification  $P$  (so that, in particular, such a process exists; uniqueness is straightforward). It is sufficient to show that the stopping time  $\tau(1)$  is almost surely finite. In fact,  $-\tau(1)$  is stochastically upper-bounded three times by a geometric variable of parameter  $3/(2\sqrt{2}) - 1$ . To simplify notations, we write  $H_t = H_t^{-1}$  and  $\underline{0} = (\dots, 0, 0) \in G^{-\mathbb{N}^+}$ . For every  $t < -2$ , if  $U_{t-1} \leq 1 - 1/\sqrt{2}$ , if  $U_t > 1 - 1/\sqrt{2}$ , and if  $U_{t+1} \leq 1 - 1/\sqrt{2}$ , then, for every  $\underline{w} \in G^{-\mathbb{N}^+}$ , we can see that

- $U_{t+1} \leq 1 - 1/\sqrt{2}$  implies that  $f_{t+1}(\underline{w}1) = \underline{w}10$  and  $H_{t+1}(\underline{w}1) = H_{t+2}(\underline{0})$ ;
- $U_t > 1 - 1/\sqrt{2}$  implies that  $f_t(\underline{w}01) = \underline{w}011$  and  $H_t(\underline{w}01) = H_{t+1}(\underline{w}011) = H_{t+2}(\underline{0})$ , whereas  $f_t(\underline{w}0) = \underline{w}01$  and  $H_t(\underline{w}0) = H_{t+1}(\underline{w}01) = H_{t+2}(\underline{0})$ ;
- $U_{t-1} \leq 1 - 1/\sqrt{2}$  implies that  $f_{t-1}(\underline{w}1) = \underline{w}10$  and  $f_{t-1}(\underline{w}0) = \underline{w}01$ , so that  $H_{t-1}(\underline{w}0) = H_{t-1}(\underline{w}1) = H_{t+2}(\underline{0})$  and  $\tau \geq t - 1$ .

For every negative integer  $k$ , let  $E_k = \{U_{3k-1} \leq 1 - 1/\sqrt{2}\} \cap \{U_{3k} > 1 - 1/\sqrt{2}\} \cap \{U_{3k+1} \leq 1 - 1/\sqrt{2}\}$ . The events  $(E_k)_{k < 0}$  are independent and of probability  $3/(2\sqrt{2}) - 1$ , which gives the result.

Thus, the algorithm converges fast. However, the dictionaries involved in the simulation can be very large. In fact, it is easy to see that the depth  $X_t$  of  $D(U_t)$  has no expectation:  $\mathbb{P}(X_t \geq k) = 1/\sqrt{k}$ . Of course, because of the very special shape of  $D_t^n$ , ad hoc modifications of the algorithm would allow us to easily draw arbitrary long samples with low computational complexity. Moreover, the

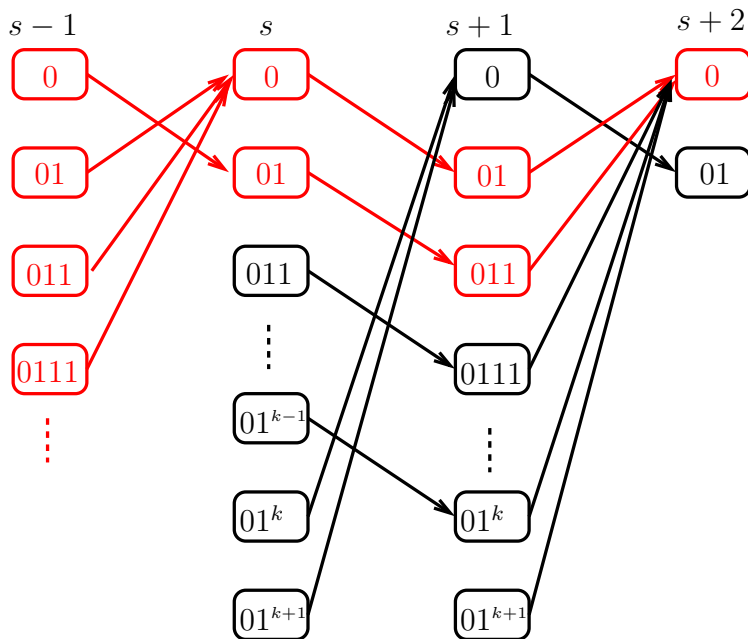


Figure 8: Convergence of the simulation scheme.

paths of the renewal processes can be simulated directly. Nevertheless, this example illustrates the weakness of the conditions required by Algorithm 1; it also shows that neither regeneration nor a rapid decrease of the coupling coefficients are necessary conditions for perfect simulation. It is easy to imagine more complicated variants for which no other sampling method is currently known.

To conclude, a simple modification of this example shows that continuity is absolutely not necessary to ensure convergence because the proof also applies to any kernel  $P'$ , such that  $P'(0|01^k) = 1 - 1/\sqrt{k}$  for  $0 \leq k \leq 2$ , and  $P'(0|\underline{w}11) \geq 1 - 1/\sqrt{2}$  for any  $\underline{w} \in G^{-\mathbb{N}^+}$ . Gallo, who has studied such a phenomenon in [9], gives sufficient conditions on the shape of the trees (together with bounds on transition probabilities) to ensure convergence of his coupling scheme. His approach is quite different and does not cover the examples presented here.

## Acknowledgments

We would like to thank the reviewers for their help in improving the redaction of this paper and for pointing out Kendall's 'coupling from and into the past' method (named by Wilson). We further thank Sandro Gallo, Antonio Galves, and Florencia Leonardi (NumeC, Sao Paulo) for the stimulating discussions on chains of infinite memory. This work received support from USP-COFECUB (grant 2009.1.820.45.8).

## References

- [1] G. Bejerano and G. Yona. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, 17(1):23–43, 2001.
- [2] Henry Berbee. Chains with infinite connections: uniqueness and Markov representation. *Probab. Theory Related Fields*, 76(2):243–253, 1987.
- [3] P. Bühlmann and A. J. Wyner. Variable length Markov chains. *Ann. Statist.*, 27:480–513, 1999.
- [4] J. R. Busch, P. A. Ferrari, A. G. Flesia, R. Fraiman, S. P. Grynberg, and F. Leonardi. Testing statistical hypothesis on random trees and applications to the protein classification problem. *Annals of applied statistics*, 3(2), 2009.
- [5] Francis Comets, Roberto Fernández, and Pablo A. Ferrari. Processes with long memory: regenerative construction and perfect simulation. *Ann. Appl. Probab.*, 12(3):921–943, 2002.
- [6] I. Csiszár and Z. Talata. Context tree estimation for not necessarily finite memory processes, via BIC and MDL. *IEEE Trans. Inform. Theory*, 52(3):1007–1016, 2006.
- [7] Emilio De Santis and Mauro Piccioni. Backward coalescence times for perfect simulation of chains with infinite memory. *J. Appl. Probab.*, 49(2):319–337, 2012.
- [8] S. G. Foss, R. L. Tweedie, and J. N. Corcoran. Simulating the invariant measures of Markov chains using backward coupling at regeneration times. *Probab. Engrg. Inform. Sci.*, 12(3):303–320, 1998.
- [9] Sandro Gallo. Chains with unbounded variable length memory: perfect simulation and visible regeneration scheme. *J. Appl. Probab.*, 43(3):735–759, 2011.
- [10] A. Galves, C. Galves, J. Garcia, N.L. Garcia, and F. Leonardi. Context tree selection and linguistic rhythm retrieval from written texts. *ArXiv: 0902.3619*, pages 1–25, 2010.
- [11] A. Garivier. Consistency of the unlimited BIC context tree estimator. *IEEE Trans. Inform. Theory*, 52(10):4630–4635, 2006.
- [12] Olle Häggström. *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2002.
- [13] Theodore E. Harris. On chains of infinite order. *Pacific J. Math.*, 5:707–724, 1955.
- [14] Mark Huber. Fast perfect sampling from linear extensions. *Discrete Math.*, 306(4):420–428, 2006.
- [15] Wilfrid S. Kendall. Perfect simulation for the area-interaction point process. In *Probability towards 2000 (New York, 1995)*, volume 128 of *Lecture Notes in Statist.*, pages 218–234. Springer, New York, 1998.

- [16] S. P. Lalley. Regenerative representation for one-dimensional Gibbs states. *Ann. Probab.*, 14(4):1262–1271, 1986.
- [17] S. P. Lalley. Regeneration in one-dimensional Gibbs states and chains with complete connections. *Resenhas*, 4(3):249–281, 2000.
- [18] D. J. Murdoch and P. J. Green. Exact sampling from a continuous state space. *Scand. J. Statist.*, 25(3):483–502, 1998.
- [19] Octav Onicescu and Gheorghe Mihoc. Sur les chaînes de variables statistiques. *Bull. Sci. Math.*, 59:174–192, 1935.
- [20] Octav Onicescu and Gheorghe Mihoc. Sur les chaînes statistiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 200:511–512, 1935.
- [21] James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995)*, volume 9, pages 223–252, 1996.
- [22] J. Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664, 1983.
- [23] Walter R. Rudin. *Principles of Mathematical Analysis, Third Edition*. McGrawHill, 1976.
- [24] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory*, 41(3):653–664, 1995.
- [25] David Bruce Wilson. How to couple from the past using a read-once source of randomness. *Random Structures Algorithms*, 16(1):85–113, 2000.

## Appendix

### Proof of Lemma 1

Let  $u \in [0, 1[$ . The uniform continuity of  $P$  implies that there exists  $\epsilon$  such that, if  $\delta(\underline{w}, \underline{z}) \leq \epsilon$ , then  $|P(\cdot|\underline{w}) - P(\cdot|\underline{z})|_{TV} < (1 - u)/|G|$ . However, Equation (1) implies that  $\phi_P(u, \underline{w}) = \phi_P(u, \underline{z})$ .

## Proof of Proposition 1

For the upper-bound, observe that

$$\begin{aligned}
\eta_P(s) &= \sup \{ |P(\cdot|\underline{w}) - P(\cdot|\underline{z})|_{TV} : \underline{w}, \underline{z} \in \mathcal{T}(s) \} \\
&= \sup \left\{ 1 - \sum_{a \in G} P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s) \right\} \\
&= 1 - \inf \left\{ \sum_{a \in G} P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s) \right\} \\
&\leq 1 - \sum_{a \in G} \inf \{ P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s) \} \\
&= 1 - \sum_{a \in G} \inf \{ P(a|\underline{w}) : \underline{w} \in \mathcal{T}(s) \} \\
&= 1 - A_{|s|}(s).
\end{aligned}$$

For the lower-bound, let  $\epsilon > 0$ , let  $\underline{w} \in \mathcal{T}(s)$  and  $b \in G$  be such that, for all  $\underline{z} \in \mathcal{T}(s)$  and for all  $a \in G$ ,  $P(a|\underline{z}) \geq P(b|\underline{w}) - \epsilon$ . Then, for all  $z \in G^{-\mathbb{N}^+}$  and all  $a \neq b$ ,  $P(a|\underline{z}) \geq P(a|\underline{w}) - \eta_P(s)$ . Thus,

$$\begin{aligned}
A_{|s|}(s) &= \sum_{a \in G} P(a|\underline{w}) + \inf \{ P(a|\underline{z}) - P(a|\underline{w}) : \underline{z} \in \mathcal{T}(s) \} \\
&\geq 1 + \inf \{ P(b|\underline{z}) - P(b|\underline{w}) : \underline{z} \in \mathcal{T}(s) \} + \sum_{a \neq b} \inf \{ P(a|\underline{z}) - P(a|\underline{w}) : \underline{z} \in \mathcal{T}(s) \} \\
&= 1 - \epsilon - (|G| - 1)\eta_P(s).
\end{aligned}$$

Because  $\epsilon$  is arbitrary, the result follows.

## Proof of Proposition 2

The equivalence of (i) and (ii) is obvious by definition. The equivalence with (iii) is a simple consequence of Proposition 1. Similarly, (iii) follows from (i): if  $P$  is continuous on the compact set  $G^{-\mathbb{N}^+}$ , then it is uniformly continuous, and

$$\varphi(k) = \sup_{s \in G^{-k}} \eta_P(s) \rightarrow 0$$

as  $k$  goes to infinity. But, by Proposition 1,  $A_k^- \geq 1 - |G|\varphi(k)$ . Finally, (iii) implies (ii).

The equivalence of (ii) and (iii) can also be proved as a consequence of Dini's theorem (see[23], Theorem 7.13, page 150): based on the definition  $\tilde{A}_k(\underline{w}) = A_k(w_{-k:-1})$ , the sequence  $\left( \tilde{A}_k \right)_k$  is an increasing sequence of continuous functions that converges pointwise to the (continuous) constant function 1, thus the convergence is uniform.

### Proof of Proposition 3

Let  $m$  (resp.  $M$ ) be the minimal (resp. maximal) element of  $G$ . Then, for any integer  $k$ ,  $\alpha_k(m|w_{-k:-1}) = A_{k-1}(w_{-k+1:-1})$ ,  $\beta_k(M|w_{-k:-1}) = A_k(w_{-k:-1})$ , and

$$[A_{k-1}(w_{-k+1:-1}), A_k(w_{-k:-1})] = \bigsqcup_{g \in G} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})].$$

But  $A_{-1}(\varepsilon) = 0$ , and thus the result follows from the continuity assumption:  $A_k(w_{-k:-1}) \rightarrow 1$  when  $k$  goes to infinity.

### Proof of Proposition 4

We need to prove that if  $U \sim \mathcal{U}([0, 1])$ , then for all  $\underline{w} \in G^{-\mathbb{N}^+}$  the random variable  $\phi_P(U, \underline{w})$  has a distribution  $P(\cdot|\underline{w})$ . It is sufficient to prove that, for all  $g \in G$ ,

$$\sum_{l=0}^{\infty} \beta_l(g|w_{-l:-1}) - \alpha_l(g|w_{-l:-1}) = P(g|\underline{w}).$$

For any integer  $k$ , it holds that

$$\begin{aligned} \sum_{l=0}^k \beta_l(g|w_{-l:-1}) - \alpha_l(g|w_{-l:-1}) &= \sum_{l=0}^k a_l(g|w_{-l:-1}) - a_{l-1}(g|w_{-l+1:-1}) \\ &= a_k(g|w_{-k:-1}). \end{aligned}$$

As an increasing sequence upper-bounded by  $P(g|\underline{w})$ ,  $a_k(g|w_{-k:-1})$  has a limit  $Q(g|\underline{w}) \leq P(g|\underline{w})$  when  $k$  tends to infinity. By continuity,

$$\begin{aligned} \sum_{g \in G} Q(g|\underline{w}) &= \sum_{g \in G} \lim_{k \rightarrow \infty} a_k(g|w_{-k:-1}) \\ &= \lim_{k \rightarrow \infty} \sum_{g \in G} a_k(g|w_{-k:-1}) = \lim_{k \rightarrow \infty} A_k(w_{-k:-1}) = 1, \end{aligned}$$

and, because  $\sum_{g \in G} P(g|\underline{w}) = 1$ , this implies that, for all  $g \in G$ ,  $Q(g|\underline{w}) = P(g|\underline{w})$ .

The last part of the proposition is immediate: for  $\underline{w}$  and  $\underline{z} \in \mathcal{T}(s)$  and for all  $k \leq |s|$ ,  $w_{-k:-1} = z_{-k:-1}$ , and

$$\begin{aligned} \bigsqcup_{g \in G, k \leq |s|} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})] \\ = \bigsqcup_{g \in G, k \leq |s|} [\alpha_k(g|z_{-k:-1}), \beta_k(g|z_{-k:-1})] = [0, A_{|s|}(s)]. \end{aligned}$$